An Intelligent Training Management System (ITMS)

Richard Stottler

Final Technical Report

January 28, 2000

Stottler Henke Associates, Inc. (SHAI)
1660 So. Amphlett Blvd., Suite 350
San Mateo, CA 94402
(650) 655-7242

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
January 28, 2000

3. REPORT TYPE AND DATES COVERED
Final Report

4. TITLE AND SUBTITLE
An Intelligent Training Management System (ITMS)

5. FUNDING NUMBERS
F41624-99-C-6022

6. AUTHORS
Richard Stottler

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Stottler Henke Associates, Inc.
1660 South Amphlett Blvd., Suite 350
San Mateo, CA 94402

8. PERFORMING ORGANIZATION REPORT NUMBER
No. 194: ITMS Final Report

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory/AFRL/HEAI
Further Marked For: Terresa E. Jackson
2504 Gillingham Dr., Bldg. 170, Suite 25
Brooks AFB, TX. 78235-5100

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
NONE

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Report developed under SBIR contract for topic number AF99-097. Air Force training units are in extreme need of advanced, intelligent training management systems to aid the training managers and schedulers in the performance of their duties and to help students quickly advance in their careers and meet the training requirements. The intelligent training management system (ITMS), to be implemented and used in Phase II, will address tracking, evaluation, requirements identification, scheduling, and completion and certification management of individuals and teams. The ITMS will perform the functions that a person dedicated to managing the training of a small group of students would perform, but do it automatically. It will intelligently guide the students as to their training needs and opportunities and help with the development, delivery, scheduling, and evaluation of courses and other training events. The ITMS intelligently models the skills and knowledge mastered by the student and makes intelligent proactive decisions and notifications based on that model. It also provides intelligent courseware tracking, evaluation, and configuration control. After determining training requirements, it intelligently schedules the required resources. The ITMS is a general tool which can be easily customized to specific domains by end-users. Several sites will use the ITMS operationally in Phase II to provide feedback and a basis for follow-on commercialization.

14. SUBJECT TERMS
SBIR Report, Training Management, Intelligent Scheduling, Artificial Intelligence, Team Training,

Intelligent Decision-making, Data-Mining, Intelligent Training Management, Student Modeling

15. NUMBER OF PAGES
83

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNLIMITED

Table of Contents

1.0 Summary.....4
2.0 Problem Description.....5
3.0 Solution Overview.....11
4.0 System Description.....14
4.1 ITMS Architecture.....14
4.2 ITMS Design.....15
4.3 Functionality.....23
4.4 Innovations.....26
5.0 Existing Training Management Systems.....27
6.0 Future Work.....28
6.1 Phase II.....28
6.2 Potential Applications.....28
7.0 References.....31
Appendixes
Appendix A. Phase I Prototype Design.....33
A.1 ITMS Architecture.....33
A.2 ITMS Skill Graph Structure.....34
A.3 ITMS GUI.....34
A.4 ITMS classes and data structures (lisp application).....36
Appendix B. Phase I Prototype User Guide.....41
B.1 ITMS GUI Guide.....41
B.2 ITMS Web Interface Guide.....43
Appendix C. Phase I Prototype Demonstration Sequence.....46
C.1 Installation instructions.....46
C.2 Notes.....46
C.3 Webpage outside firewall: http://207.214.202.162/cgi-bin/main.cgi.....46
Appendix D. Demonstration Sequence (Screen Dumps).....49

1.0 Summary

In our Phase I project we accomplished all of the objectives listed in the Phase I proposal. Perhaps most important was the fact that we proved the feasibility of the concepts described throughout this final report in a prototype, which implemented functionality in most of the areas envisioned by the Phase II design included herein. For example, it had an E-mail interface and could send students proactive notifications through E-mail. It modeled the students general and specific skills using a skill hierarchy which could be edited by users and included both the more specific and subtask relationships. Users could edit the course descriptions, which included different versions for the same course. The course descriptions included prerequisite skills required and the skills developed (learning objectives) by the course along with minimum computer requirements.

The ITMS ran on a web server and wrote relevant information to the appropriate web pages. When a new version was created, a student who had taken the old version would be notified both through E-mail and through a web page tailored to the specific student. The prototype ITMS would notify students when they we're falling behind or when their skills had decayed. It would automatically E-mail them questionnaires after they took a course and expect a response in a reasonable amount of time. This time expired the student was reminded until eventually his supervisor was notified via E-mail.

Users could edit job descriptions and edit the career map, graphically specifying prerequisite relationships. This information was used to provide career counseling to students on their own web page. This included determining if the goals were realistic and determining what course the student needed to take and in what order. If the student's existing skills didn't meet those required as prerequisites for a course, the prototype would search for additional courses that the student was eligible for that would allow him to reach the required skill levels.

The prototype even had permission management capabilities. Students or instructors could be individually authorized with passwords and each had privileges appropriate to their class. The prototype would also evaluate courses based on the data that it had from students, supervisors, and the course results. This was a simple version of the algorithms described in Section 3.

The design for the phase II system, and therefore for the prototype, whose primarily role was to prove that design's feasibility, was based on interviews with operational experts in the training management process. Our first discussions were with various Air Force officers with AWACS team training management responsibilities at Tinker AFB in Oklahoma. We also received and analyzed documents relating to the complex training requirements relating to aircrew (especially pilot) training. We also had discussions with the several individuals with training management responsibilities at the Army's military intelligence distance learning center at Fort Huachuca, Arizona. Finally we confirmed that the Navy's needs paralleled the Air Force's and Army's through discussions with Commander Pinto, XO of the USS Paul Hamilton.

2.0 Problem Description

The Department of Defense' (DOD's) training requirements are changing, primarily because the jobs that DOD personnel do are changing. More thinking is required of all military personnel at all levels, primarily problem-solving – thinking through difficult problems. These changes are a result of "the new world order." That is, with the end of the cold war, the US faces asymmetric threats (enemies with far less capability than itself). Low intensity conflicts and military operations other than war (MOOTW) are more the norm than the exception. These lead to unpredictable situations and ill-structured problems. These circumstances require a higher degree of flexibility in the individuals.

Additionally, there is a much greater number of these asymmetric threats. During the Cold War we faced one, or possibly a few, credible threats with known doctrine. This could be studied and tactics developed, in advance, to counter likely enemy actions. But now, with hundreds or thousands of possible threats, many only vaguely known or even completely unknown, it is impossible to study or understand all the possible adversaries, their capabilities, doctrine, and tactics. Thus, it is impossible to design appropriate responses to their actions in advance and train our military personnel in those actions. Given the unknown nature and behavior of the threats, cognitive flexibility of our forces is paramount.

The world continues to rapidly evolve in other ways as well. Equipment rapidly changes – both those in use by our own forces and those used by potential threat forces. New software versions used by our forces may be updated multiple times per year. It is impractical to retrain our military personnel in the specific capabilities of a new equipment (either equipment of ours or possessed by potential threat forces) every time it changes. Rather our personnel need to have the general abilities to adapt to new equipment, without retraining each time. Their original training should not be for specific equipment, but rather how to understand the new capabilities, on their own.

Because the requirements of military jobs have changed, the training for those jobs must change. For example, there is already less emphasis on training specific procedures, preprogrammed reactions, doctrinal rules, rote memorization, and behaviorist training approaches. More emphasis is already being placed on training in the context of scenarios. That is, training is being conducted by practicing for a variety of situations. There are several theories of learning that relate to training with scenarios. These include Situated learning, Anchored instruction, Scenario-based training, Simulation-based learning, Case-Based instruction, constructivist theory, and several others. Many of these place explicit emphasis on the notion of training more abstract, general, problem solving skills and less emphasis on specific, concrete procedures and tasks. Because these training strategies embody more complex methods of instruction, and because there is a greater emphasis on more general (and subtle) skills, more complex tracking of student training is required. It is not enough to simply know which student took which course, because two different students may have had widely varying experience in the same course, and evaluating and tracking general problem skills requires more sophistication. It is not enough to simply track which course a student took or even which of his responses were correct and incorrect. Instead a system must evaluate and track his mastery of both specific and general skills and knowledge.

It should be noted that we have a very general definition for the term "course" as used herein. A course is any defined learning experience or training event and includes Distance Learning, correspondence courses, residence courses, on-site training, training sorties, simulator training, on-the-job training, etc.

The DOD training situation is also changing. Training budgets are being radically reduced. This is forcing training units to radically reduce or eliminate school house course lengths. This training is being replaced with distance learning alternatives, either as optional training or as explicit prerequisites for other courses, jobs, or promotions. The new courseware being produced to fill this void varies widely in terms of level of sophistication and pedagogy, depending on who is developing it.

The DOD also has very unique training (and training management) requirements. The training is often life-critical with complex, time constrained, high-value decision-making. The training requirements regulations are very complex. There is a need to provide just-in-time training which is specific to a particular mission or geography. Finally, the DOD has begun to appreciate the importance of managing the training of its personnel across their entire career – the Life Long Learning concept. This leads to very long student tracking times.

Many of these problems specific to the DOD are exasperated by the specific needs of Air Force aircrew team training. Air Force training units are in extreme need of advanced, intelligent training management systems. As warfare has gotten more complex (technology and tactics), Air Force training requirements have increased. But at the same time, Air Force training budgets have been reduced. This is true for both initial and sustainment training of active duty, guard, and reserve personnel. Training which includes academic schools, simulators, flight, on-the-job training and distance learning courseware, are all affected and in need of a new generation of training management systems.

The reduced budgets and increased requirements have forced training units to do more with fewer resources. Optimal utilization of these scarce resources has become more important. This has significantly increased the importance, complexity, and difficulty of scheduling training events. Even after a good schedule is developed, it is subject to many dynamic events and constant rescheduling is the norm. This occurs for several reasons including weather (not acceptable for training requirements), lost resources (equipment breaks or is re-allocated), trainees becoming unavailable, etc.

Determining the training requirements for individuals is also complex. This is because the regulations governing training are so complex and any one individual is subject to many different regulations. For example, some training, such as small arms training, is required of all Air Force Personnel. Other training requirements apply to all air crews, such as altitude chamber training. Additionally, pilots (who are also subject to the requirements for air crews and all personnel) must stay current on the relevant air platforms or they will require refresher training. As this example shows, any particular individual may be subject to many separate requirements. Furthermore, many of these regulations have complex periodicity rules. For example, a pilot may be required to log a certain number of flight hours each month, without too great a period of time transpiring between flights. Missing the month target may then activate the 90 day minimum requirements. Depending on the degree of the deficit, there will be different training requirements to make the pilot current again.

With each individual having to meet so many and such a complex array of requirements, tracking which requirements have actually been met is also difficult. Not every student attends every event for which he is scheduled. Thus, the "as-attended" record of the training events must be used.

These problems are further exacerbated for the many team training domains. An AWACS aircrew has about 18 positions. Thus, 18 different trainees have to be scheduled for most AWACS training events. Most of these fall into different categories, each with its own unique training requirements. Additionally, several different types of instructors have to be scheduled, who have their own training requirements to meet in order to be acceptable as instructors and to be allowed in the air. In the AWACS domain, non-AWACS aircraft, which are under control of different units altogether, must also be scheduled for most airborne training events. These aircraft may or may not have symmetric training requirements. For example, while tanker aircrews have training requirements relating to practicing with AWACS crews, and are thus relatively easy to negotiate schedules with, fighters have no such requirement. That is, while the AWACS Weapons Directors (WDs) have a training requirement to control fighters in various airborne scenarios, the fighters have no corresponding training requirement to be controlled. They can be very difficult to schedule.

These training management problems are especially difficult for reservist and guard units, whose members have full-time jobs. They are not available every day. Furthermore, on occasion, something happens at the trainee's job which prevents him from making his scheduled Air Force training event, sometimes with little or no notice.

We have spoken to several different individuals involved with the training management problems for various units who train out of Tinker AFB in Oklahoma. All were very informative and eager to help and provide many useful examples of current problems, some of which are described below.

An example from the reservist AWACS domain helps illustrate the training management problems. One of our contacts proposes that scheduling AWACS reservist sorties is one of the hardest training management tasks in the Air Force. First, he has to coordinate the individual schedules of the trainees manning the 18 separate positions, each of whom has another full-time job, which creates additional scheduling constraints. These jobs make the typical 8-12 hour active duty AWACS sortie length impractical. So the sortie length is generally reduced to 6 hours. Furthermore, the trainees are only available at certain times during the week. These problems are magnified by 6 since he is in charge of scheduling for 6 different teams. He then has to coordinate his training sorties with fighter training sorties in the area. The fighters fly in several different profiles, many of which generate no events, and therefore, no training for the AWACS crew. So he must piggy back his AWACS sortie onto fighter sorties who are scheduled to fly the correct types of missions. Finding these is a difficult problem since the fighters can more easily fulfill these certain profile training requirements using ground controllers endemic to their units.

Even after this complex schedule of sorties is worked out, problems often materialize. It is common for the fighters to cancel their sorties with 12 to 24 hours notice. In order to preserve the training sortie and reservists training schedules, the scheduling team then quickly scrambles to find alternate fighter training sorties. Usually they call units in a four to five hundred mile radius. They typically call fighter reserve units first. This is for two reasons. The fighter reserve units can be more understanding of the special needs and problems of other reservists. Secondly, since reservists tend to be more experienced than their active duty counterparts, they make fewer mistakes, so the fighter reservists prefer to train with AWACS reservists. About 75% of the time, they are successful in finding alternate reservist fighters to be controlled. The next fall back is to try to coordinate with active duty fighter training. The final fall back is to perform only surveillance training. That is, there are some training requirements for some of the AWACS crew that can be fulfilled by using the onboard sensors to monitor commercial air traffic. But certainly, weapons directors fulfill no requirements in this final fall back mode of training. Because of these problems and issues, maintaining the training sortie schedule for 6 AWACS crews requires 4-5 full time schedulers.

The most time-consuming aspect of their jobs relates to creating the schedule and disseminating it (and the frequent changes and updates) to the wide variety of individuals involved. In addition to the many people directly involved in the training (trainees, instructors, pilots, etc.), there are a large number of people not directly involved who also must be notified. For example, the maintenance unit, who is otherwise not involved in the training, must be notified so that they will be available and so that the equipment is available.

Another Tinker AFB contact was in charge of the training management for about 25 AWACS Weapons Director reservists. This requires about 30 hours per week. Her first problem is determining the training required for each individual. This is complicated by several factors. First, an Air Force wide automated system keeps track of each individual's flight hours. Unfortunately, there is a two step entry process, where the trainee fills out a form which must be entered by someone else. Occasionally, this is not done correctly, and the error is usually not uncovered for several months or longer. By that time, it is often difficult to remember or reconstruct which flights which trainees were on. Furthermore, as alluded to earlier, the regulations which describe what the training requirements are, are themselves complicated. Keeping track of how often different recurrent training must be done (whether quarterly, yearly, every 2 years, every 6 months, etc.) and matching those to each individual is difficult. Flying requirements are more complicated, requiring a certain number of days per month, without too long between flights and fall back requirements spanning 90 days. Again these must be applied to each individual. The result is a complicated array of training events for each trainee which includes such diverse items as life support training, chemical warfare instruction requiring special equipment, altitude chamber training, academic training, simulator training, training sorties, etc.

Once the events each trainee needs are determined, significant difficulties remain. Scheduling of training events occurs in a distributed manner. For example, the AWACS training manager (managing the training for 25 people) must coordinate with the training sortie manager (who provided the other example, above) as well as a large number of other individuals who each manage their own set of required training resources and events. A negotiation process occurs where they balance the needs and schedules of trainees against the availability and schedule of resources and related training events.

Even once the schedule is set, problems occur. The training manager discussed several types of problems. One problem, more common with reservists, is last minute cancellations, due to illness or job requirements. This requires rapid rescheduling to fill the required position, so that training can occur for the other positions, hopefully with someone who needs the training in that position. Furthermore, care must be taken to track the fact that the originally scheduled trainee did not attend the event and therefore still has an open requirement for it.

Another problem that occurred recently was that an instructor was unable to make the training flight. This resulted in a lot of last minute scrambling to find someone with the applicable training credentials to fill the spot. A system which keeps track of all training resources, including instructors, both as to their availability and their whereabouts, would greatly aid this rescheduling process.

A third contact reiterated many of the same problems and issues, but also brought up several others. It is important for senior leadership to see how far along the trainees are so that they can determine how many more sorties or how much more simulator (and other resource) time to buy. Furthermore, an automated system must be able to handle large training events and large aircrews. The system must provide Web access to allow trainees to sign up for courses and events and provide their availability information. An intelligent system must be capable of deconflicting the schedules of trainees and of the resources. One problem they currently have in particular is having trainees scheduled for two separate events at the same time.

Widely varying and rapidly changing training requirements result in there being many different versions of the same course. At a particular moment in time, there may be different versions of the course in terms of different scenarios (perhaps for different types of missions or different geographical locations) or for different types of computer hardware. Particular training organizations may need to frequently update their courses as well, leading to multiple course versions each year.

Although there are a very large number of existing training management systems, these do not begin to meet the complex needs discussed here and do not contain intelligent features. These systems are primarily networked database systems and store data relating to course catalogs, class schedules, enrollment, student information, transcripts, class evaluations, homework, self-assessments, course authoring, content management, grades/test scores, and rudimentary skills. The primary benefit they provide is that of a pre-customized DBMS with existing interfaces defined to the vendor's own courseware offerings or authoring tools. The primary disadvantages are that they do not attempt to track higher level skills and they do not exhibit intelligence, decision making, or proactivity, leaving these functions to the training managers or the students themselves.

An intelligent system is needed to help manage the complex training process. It should perform the functions that a person dedicated to managing the training of a small group of students would perform, but do it automatically. This would achieve an ideal which is rarely achieved. An Intelligent Training Management System (ITMS) would intelligently guide the students as to their training needs and opportunities and help with the development, delivery, evaluation and scheduling of courses.

The problems discussed above dictate the requirements of an ITMS. The ITMS will keep track of what general and specific skills, knowledge, and tasks the student has mastered over time. It will use that information to proactively help the student manage his career and the life-long learning process. After determining the training requirements, it will schedule the required training resources, including instructors and other team members (other students). It will track the different, changing versions of courses and help manage the change and notification process. For example, it will keep track of which student took which version of a course and know how they are different. Given relevant data, the ITMS will automatically produce an evaluation of each course. In addition to capabilities for students, instructors, and course authors, it will provide functionality for supervisors, mentors, course evaluators, and training managers. The ITMS will support the management of the permissions and authorizations to access the various data and functionalities.

The ITMS will provide intelligent student tracking and learning/career management. It will keep track of where the student is at in his career, in terms of what courses and jobs he's had. But more importantly, it will keep track of what skills, knowledge, and tasks he's mastered over time. These will include general and abstract skills, not just specific, concrete ones. For example, one new skill is the ability to learn about new equipment and how to troubleshoot it. Another is the ability to adapt to new enemy capabilities. When tracking a student over a long period of time, many things can change. His job requirements change. The courses change. Some of his skills decay from lack of use. (That is, skill mastery doesn't always increase). Because the ITMS is tracking these skills over a student's entire career, and because the required skills change frequently, the ITMS must allow the training managers to update the general and specific skills taught by courses and required by jobs. The ITMS must be able to track prerequisites taken by the student and required by him for future events. The ITMS will be able to calculate these prerequisites, even given the complexities of determining them in the face of complex regulations and skill requirements.

The ITMS will be proactive – telling students what prerequisites they need to finish before taking courses, nudging them when they fall behind, and informing them of possible skill decay. This will occur when the student is only using part of the skills for which he was trained, in his current job. This is especially important if his next job assignment will be using a different set of skills. In that case, it should evaluate those skills (with a no-penalty test) and remediate the student with refresher training as appropriate. The ITMS will inform students of the need for updated knowledge and skills for their jobs and new courses or new versions of old courses that address those deficits.

The ITMS must also address individuals and teams. The ITMS must be able to independently make decisions and recommendations but also accept input and overrides from training personnel. As part of determining the training requirements and tracking their completion, the ITMS could also evaluate the results of the various training events, either for individuals or teams. This would allow it to be able to automatically schedule additional, or remedial, training as required. Another issue which an ITMS could easily address is that of deployment. When a team must be quickly assembled to deploy, the ITMS has all of the skill, training and availability information to select the best team, either as a whole or assembled from individuals. It can also identify the additional training required of a team and its members to meet the needs of a particular deployment.

3.0 Solution Overview

The Intelligent Training Management System's primary focus is the student and its primary objectives are to maximize his efficient training and to further his career development in the context of life-long learning and general problem-solving. The tools it has available to it with which to accomplish these objectives are primarily the different types of learning opportunities and training events, and, evaluation methods, although all of these are constantly changing. These include distance learning, on-site, and correspondence courses; on-the-job-training; tests, just-in-time scenarios, simulator training, training sorties, etc. In order to make decisions regarding its actions, the ITMS has several types of knowledge available to it, including prerequisites, course learning objectives (which skills are taught by the course), training requirements regulations, job descriptions (which skills are required and practiced by various jobs), estimates of the decay rate for those skills, available resources, and the career map which describes the progression and prerequisite relationships between courses, ranks, and jobs. All of these change over time. The ITMS also has sources of additional knowledge including the student, his supervisor, course results, and evaluation results.

The ITMS will determine the applicable training requirements for trainees and teams. It will schedule the required events, including the trainees, instructors, and other needed resources. It will track the results and update the trainee's and team's histories. It will be able to select the best teams for deployment on particular missions and what training is required for a particular team to perform a particular mission.

The student tracking solution is based on the intelligent student model, borrowed from the realm of intelligent tutoring systems. The ITMS explicitly models the student's currently mastered skills, knowledge, and tasks. These are the stated and more general learning objectives of courses. They are organized by the instructor as a multiple dimensional hierarchy primarily around the more-general and subtask relationships. The student model also includes a description of which jobs the student has had and which courses he has taken. Since both are subject to change over time, the student model actually references specific versions of each. The courses and job descriptions utilize the same vocabulary (the hierarchy of skills, knowledge, and tasks) used by the student model. These are used to infer mastery levels in the student model. The ITMS will, when appropriate, automatically question the student and his supervisor regarding the specific and general skills taught by courses, and the degree to which they are successfully taught, skills required for (or not used in) various jobs, and the student's current degree of mastery of those skills.

The student's mastery changes either up or down over time. This is modeled with estimates of the skill increase provided by courses and jobs (on-the-job training or simply practice and experience) and heuristic skill decay factors which become specialized to the student, through data mining techniques, after the ITMS had had a chance to observe him over a long period of time. The ITMS also knows how quickly the student should be completing courses and progressing in his career. The ITMS uses the student model to proactively make decisions and notifications. It also contains more mundane information such as contact information (E-mail, phone, mailing address), available computer resources, his supervisor, etc.

After training requirements for each team and individual are determined and approved, the system would attempt to schedule the applicable events. The ITMS would make use of each student's availability, constraints, and requirements to come up with the most desirable schedule for the team as a whole. It would also have to negotiate with the managers of the training events or applicable resources. This negotiation might be with the human managers, in which case they would be sent an e-mail with a form to check-off the possible available dates and capacity of the required events. Or it might be with an ITMS component, which is local to the training event or resource manager. In that case, several messages can pass back and forth as to an optimal event schedule, given the needs of the students and of the events and resources. All related training managers could alter the schedule or add additional constraints. Since we have found in most scheduling problems that there is often a large number of reasonable schedules, and since not every constraint is always defined to the scheduler, the ability to manually adjust the schedule, while the ITMS continues to check for constraint violations or resource conflicts, is very useful.

The ITMS would provide reports to senior leadership about the progress of the training and the additional resources required to meet applicable training targets. When requested, it could assemble or select the most applicable team based on mission requirements. It could also determine the additional training that is required to bring one or a set of teams up to the requirements of some particular type of mission. If requested, it could then schedule the applicable training events.

Along with the student model is an intelligent course model. It uses the same vocabulary (skills, knowledge, and tasks hierarchy) as the student model to describe its learning objectives. Because a course will have different versions (such as which scenarios were actually used for a particular student as well as due to updated content over time), each course is actually a complex web of versions and scenarios. Each version has its own (partially different) learning objectives, history and student lists. The ITMS automatically uses actual student performance to evaluate the course in terms of how well it meets its learning objectives; that is, how well it teaches the specific and general skills.

The course model is used by the ITMS as its starting point for students who have taken the associated course. ITMS's first estimate of the mastery of a skill by a student is based on the results from the course that first teaches that skill to the student. (This estimate is later updated based on supervisor evaluations, relevant on the job experience (or lack thereof), decay factors, and future learning events). The course model can handle prerequisites in one of two ways. Courses, jobs, or ranks can be explicitly listed as prerequisites for other courses jobs, or ranks in a career map. Or, the required mastery level of skills, knowledge or tasks can be listed. This latter method provides more flexibility for the ITMS in terms of how it recommends that prerequisites be fulfilled. For example, if one course is listed as an explicit prerequisite for another, the ITMS will be forced to require the student to take the first course before the second. However, if a course lists skill levels as its prerequisites, there may be multiple methods of achieving those requirements. It is even possible that the ITMS estimates that the student already has the prerequisite levels (perhaps through on-the-job experience, or other courses).

The training manager will also be able to specify rules to allow waiving of prerequisites. The ITMS will calculate their effect in terms of the number of eligible students and likely course throughput.

Given the intelligent course model, it's relatively straightforward to add version control and tracking and configuration control functionality. Version control and tracking includes keeping track of which courses and versions are available and what each teaches; which students or units have which versions, whether they were distributed via CD or the Internet; making sure students are using the correct version for their particular needs given their computer constraints; and making sure the appropriate students are notified of course updates.

Configuration control refers to aiding the courseware development process by tracking the different versions of the separate files that make up the courseware, making sure that all the development team is using the most up-to-date versions, and that the final released product is the most up-to-date version. These capabilities are important when several different individuals are involved in authoring the course.

A final ITMS capability is automatic courseware evaluation. It is facilitated by the student and course models. The ITMS evaluates the course's ability to meet stated and more general learning objectives. It will use both in-course test results (which is somewhat circular) as well as after-course test results. The ITMS will use job performance, based on questioning the supervisor on specific and general skills of the student. It will also use the student's performance in follow-on classes. It will initially evaluate the course-based results from a pre-release test class, if available. It uses the information in the student models to estimate a course's ability to impart skills, knowledge, and task mastery. Using constraint satisfaction, it can also assess the course in reference to specific students and use data-mining techniques to look for patterns to see what attributes a student needs to lead to a good or bad performance in particular courses. This is helpful feedback for the course authors since it tells them if their course is particularly good or bad for certain kinds of students, so they can take advantage of it or take the opportunity to fix it.

4.0 System Description

4.1 ITMS Architecture

The high level ITMS Architecture is given below. The ITMS resides on a web server. We have identified 8 different types of uses, each of which interacts with the ITMS primarily through a web interface customized to that type of user. Each is described further below. The ITMS updates each user's specific web page with information, which is particular to that user. Additionally, for proactive notifications, the ITMS will be interfaced to an E-mail system so that it can send E-mail to any users that have it. The ITMS will also have the capability to print out physical letters, in the event that a user is unreachable with E-mail. The ITMS will also have a database interface to receive information from external databases and update them, if required. These databases include personnel databases (to get a student's contact information, current rank and job, supervisor, etc.), registrar databases (to determine who is currently registered in what course, what future courses, what past courses, and any certifications), course results databases (to get student's course results), and course description databases.

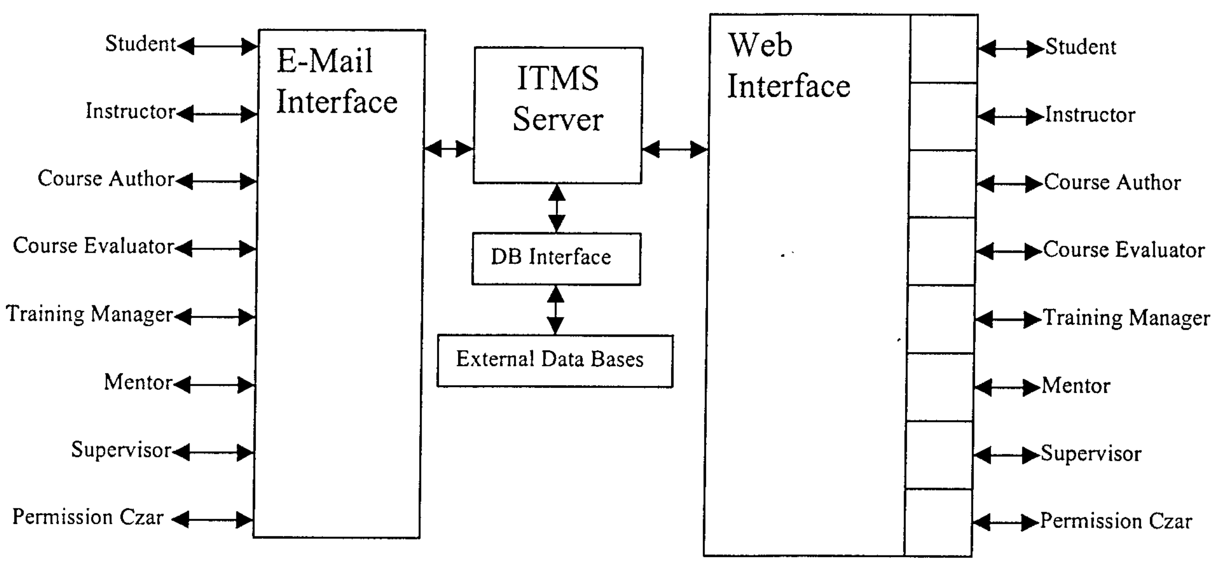


Figure 1. ITMS High Level Architecture

The student has a personalized web page that ITMS updates with information specific to him. For example, if there is a new version of a course that he has taken, that information will be posted on his web page. When the ITMS schedules him for a particular training event, that information will be E-mailed and posted to his personal web page. The student can use his web page to make queries and generally see what courses are available and what information the ITMS has deemed especially relevant to him. This is also where he keeps his contact and other personal information up-to-date, including his availability for training sorties or other hard-to-schedule events with limited resources and opportunities. If the ITMS has had trouble contacting him, it will specifically request up-to-date contact information when he logs on. The student can view his own skill mastery levels, and receive career counseling guidance as well. The proactive E-mail-type notifications include the need to take prerequisites, the fact that he is falling behind in completing those prerequisites or other requirements of his career path, updated versions of courses or knowledge required for his job, and skill decay warnings.

The instructor has authorization and capabilities through the web page to view the models of students currently in his courses, add new students to his courses, remove students from his courses, and register the student's result data for his courses. He can also view student questions or products and send answers to all the course's students, a particular student, or post them to the course web page.

The course author maintains the descriptive information for course versions through his web page. He can view the skill requirements and other prerequisites for various jobs, edit the skill and other prerequisites for his courses, edit the estimate of the specific and general skills mastery that the course accomplishes (the learning objectives), and view evaluations of the course. He can also add to the skill, knowledge, and task hierarchies. He can also access the configuration control capabilities.

The course evaluator reviews the course and inputs his review, evaluation, and suggestions, which only the course authors can view. He is also authorized to examine student models for students taking the course, but without access to their names, so he can see if his hypotheses about the strengths and weaknesses of the course are valid.

The training manager, through the web interface, can edit the skill requirements and other prerequisites for various jobs and add to the skill, knowledge, and task hierarchies. If an ontology conversion is required, perhaps because a job or its vocabulary has changed radically, he can define the mapping from the old hierarchies to the new. He can also input waiver rules and view the resulting course eligibility and projected throughput.

Particular students and/or particular courses or jobs may have designated mentors. A mentor, through the web interface, can view a student's skill mastery model and his career plan. He can answer the student's questions relating to his current job, course, or career.

A student's supervisor, through the web page, can view the student's skill mastery level, submit his own estimates of the student's abilities for both general and specific skills, and update the skills needed and practiced for the student's current job. He will also be notified if the student has not responded to the ITMS in a reasonable period of time.

The permission czar authorizes various users and classes of users to have access to the data and capabilities within the ITMS. In addition to the 8 roles described here, he can create new roles as new combinations of authorized capabilities and data access.

4.2 ITMS Design

The ITMS design is shown below. The heart of the system is the base ontology maintained by the course authors and training managers and which will be referenced by all of the models in ITMS. This base ontology will be described as multiple hierarchies of skills, knowledge and tasks (hereafter simply referenced as "skills"). These are the skills and knowledge required to perform particular jobs. The tasks are the tasks required to be performed in a particular job. These skills may be either general or specific. For example, a particular weapon director may possess the specific skill of recognizing the champagne air-to-air tactic. Another weapon director may possess the more general skill of recognizing any tactic that is attempting to outflank the opposing force. This example also illustrates one type of relationship modeled between elements in the hierarchy – the more-general (or more-specific) relationship. Under this relationship the skill of recognizing any tactic that is attempting to outflank the opposing force will have several children including recognizing the champagne tactic, recognizing the bracket tactic, recognizing the pincer tactic, etc.

The other type of relationship supported between skills is the subtask relationship. The task of a weapon director making picture calls to fighter pilots consists of detecting the enemy tracks, recognizing their tactics, determining which friendly fighters are affected, formulating the proper radio transmissions, then making them. Each of these is a subtask of the making picture calls task.

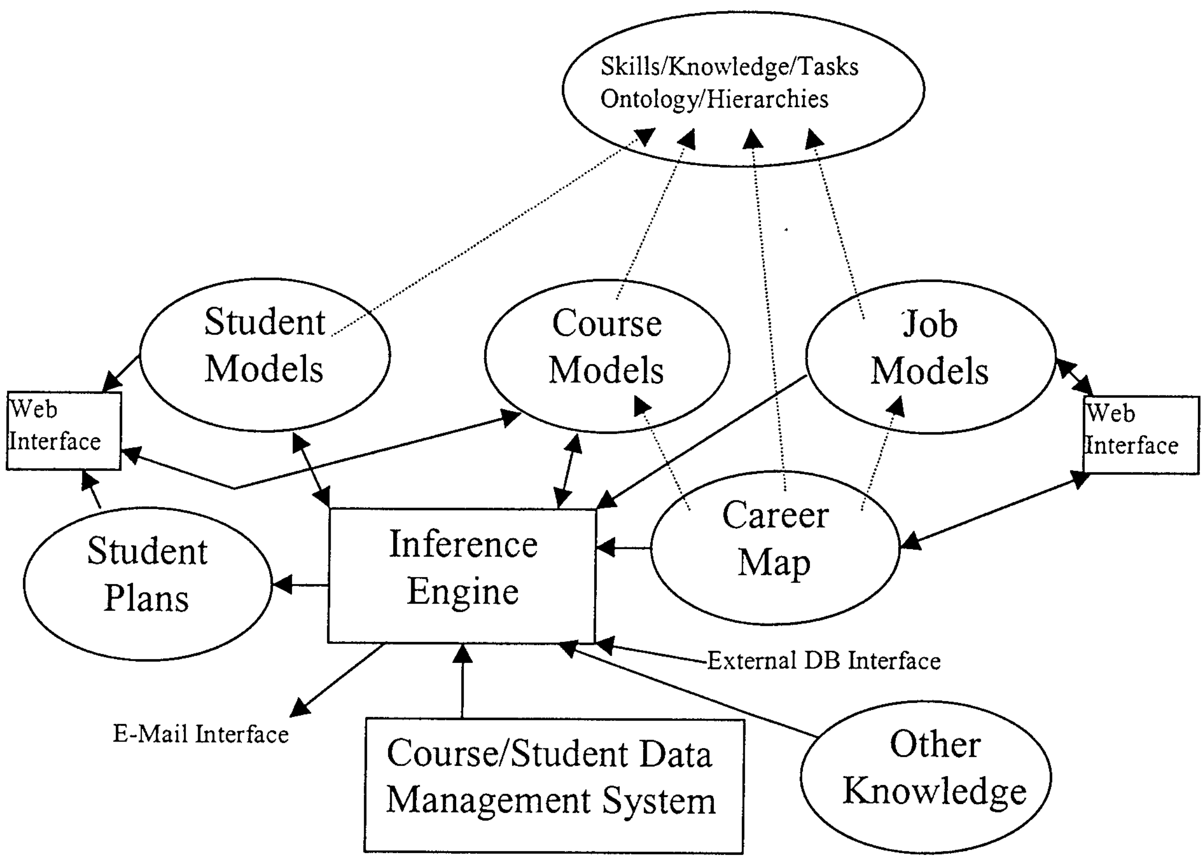


Figure 2. ITMS Design

Course Models

In the figure above, dotted line arrows connote reference links. That is, the course models, student models, job models and career map all reference the skills hierarchies. The course authors maintain the course models and decide at which point the updates are significant enough to warrant that a new version should be defined for the course. The course models include learning objectives which are lists of skills from the skill hierarchy (either general or specific or a mixture of both) as well as the degree of mastery expected from students completing the course. The course model also lists explicit prerequisites which may be any course, job, rank, or other object appearing in the career map. Required prerequisite skills needed to successfully take the course can also be described in the course model using items from the skill hierarchies and degree of mastery required. The course model, in addition to the course author's estimates, will also include ITMS's estimates of the course's ability to make student's achieve various levels of mastery. These are statistical estimates based on constraint satisfaction applied by the inferencing engine. The course model will include estimates from course evaluators as well. The course model (for editing and examination of ITMS's quality estimates) is available to course authors through their web interface.

Job Models

The job models are maintained by training managers and, indirectly, by supervisors. Each job consists of a list of skills from the skill hierarchy (either general or specific), as well as the degree of mastery required to perform the job. If skills are expected to improve during the course of the job or other on-the-job-training is expected to occur, the mastery level expected at the end of the job assignment is also described. These initial estimates are also updated by ITMS as it gathers more data, primarily from supervisors of the students holding the relevant jobs.

Career Map

The career map primarily shows the relationship between the various courses, jobs, and ranks in the domain. Arrows between these objects represent explicit prerequisite relationships. For example, a particular rank may be required to take a particular course, which is required before assignment to a particular job. In the career map, prerequisite arrows would be shown from the rank to the course to the job. Any object may have any number of explicit prerequisites and may be the explicit prerequisite to any number of other objects. Objects can also be implicit prerequisites for each other, by the definition of prerequisite skills described above. The course map objects also include heuristic knowledge as to how fast they can be expected to be accomplished. The career map can be accessed by students who are in the process of determining their career goals and is maintained by a manager for that specific domain. For example, the career map for AWACS weapons directors (WDs) would be maintained by the manager responsible for defining the requirements and prerequisites for AWACS weapon directors and senior director jobs.

Student Models

The student models are generated by the ITMS and basically copy the structure of the skills hierarchy. For example, the student model for a particular AWACS WD would contain the entire hierarchy for the AWACS WD domain, both the low-level, specific skills and the high-level, more general skills. For each skill in the hierarchy, the ITMS will have estimated the mastery of the particular WD in that low or high level skill. The first estimates for a skill are based on the first course or job that develops some mastery in that skill. This estimate is refined with additional data as it becomes available to the ITMS over time. Furthermore, the ITMS will make additional inferences as appropriate. For example, if all of the subtasks for a task are mastered, then it is likely the task itself has been mastered (subject to the ability and need to perform them concurrently). Similarly, if a course teaches the general ability to recognize tactics and, additionally, the student has demonstrated some proficiency in recognizing specific tactics, it can be inferred that he can recognize a variety of tactics (or easily learn to), even if he has not been tested on them before.

Course/Student Data Management System

The ITMS will be able to track and manage thousands or even millions of students. Thus, the ITMS will store the information associated with students and the student model in a database management system (DBMS). Similarly, there will be a lot of information associated with the results of particular students taking particular courses and these will also be stored in a DBMS. This is indicated in the design by the "Course/Student Data Management System" box.

The ITMS needs to get the results of each student taking each course. In the case of resident courses, this would likely occur using the interface to external databases, so that a batch of students who have just completed a course, and whose results are stored in a database, could have their results input electronically, all at once. But in the case of electronic courseware, it would be best to have the courseware automatically send the results to ITMS. SHAI will provide a library of code which course authors can use and easily add to their courseware so that the results are sent back to ITMS when the student completes the course.

Similarly ITMS expects to get results in the vocabulary of the skill hierarchy. The courseware may have the data in a different form, such as listing of correct and incorrect student actions. SHAI has developed existing code which can be customized by course authors to convert student action lists to estimates of the degree of mastery of skills. This code will be packaged and provided to the course authors as well.

Student Plans

The student plans are generated by the ITMS in consultation with the student. The student examines the career map and selects career goals from the objects in it. ITMS then examines his current accomplishments (in terms of mastered skills, courses taken, and jobs and ranks held) and computes what is required, in what order, in how long it is likely to take, and reports this information back to the student.

Other Knowledge

In addition to the models and career maps, there is a large amount of other significant knowledge which can be edited and input into ITMS by users. This includes heuristic a priori decay factors; contact methods with heuristic estimates as to their success probability, likely delivery times and level of effort; proactivity knowledge and training regulations and requirements knowledge, as described below.

One specific type of knowledge represented with ITMS is general knowledge of training requirements for individuals and teams. ITMS includes an intelligent training requirements module which uses this knowledge to determine the training requirements for each individual and team. We use an object-oriented approach to represent training regulations and requirements. Within the ITMS, objects exist which correspond to different positions and teams within the Air Force and which describe the applicable training requirements. Through a multiple inheritance scheme, objects which correspond to individual trainees are dynamically instantiated as members of the applicable classes. For example, an AWACS pilot would be instantiated as a member of the following classes: Air Force personnel, air crew member, pilot, and AWACS pilot. Furthermore, the pilot's team would be instantiated as an air crew and AWACS team. Associated with each class are the data and methods to calculate the training requirements, based on the individual's and team's histories. These requirements are themselves objects that describe the resources needed for the training requirement, or event. The training event might simply be 20 hours of classroom training on applicable threats. In that case, the resources might only be a classroom, instructor, and presentation materials and devices. A more complex training event might be a Defensive Counter Air (DCA) AWACS sortie. The required resources would be an AWACS aircraft, 18 trainees to man the positions, perhaps 10 instructors, a tanker and crew, 4 F-16s to be controlled (along with their pilots), several aircraft and pilots to fly the threat profiles, and enough airspace to play the engagement.

The requirement objects for each individual and team are displayed in the Identified Requirements Editor. These can be accepted or supplemented by the training manager. These are sent to the intelligent scheduler, possibly with the addition of more constraints. ITMS includes an intelligent scheduling capability that manages all the resources under the corresponding training manager's control. Different types of training managers have different types of resources under their control and so the different intelligent schedulers perform different functions, but all will use the same underlying technology.

Intelligent Scheduler

The following example illustrates the complex negotiations automatically managed by ITMS. A training manager in charge of the 25 AWACS weapons director (WD) trainees, is primarily tasked with determining the training requirements for each and managing their schedules. The intelligent scheduler on her local PC is only free to select times for the trainees (subject to their entered availabilities and constraints) but not to schedule/allocate resources under someone else's control. If it was determined that several WDs required a training sortie, a request with their available dates would be sent to the intelligent scheduler residing within the ITMS on the AWACS sortie manager's desk. This scheduler, while having no control of trainees, would have control of the resources controlled by the sortie manager, including AWACS aircraft and maintenance crews. Unfortunately, it does not have control of the required fighters or their pilots and so would have to send requests for those assets to applicable fighter units. If those units were also running ITMS, their intelligent schedulers could automatically respond to the requests. If not, E-mail requests would be sent to the managers of the fighter sorties requesting specific dates and times and/or what existing fighter sorties could be piggy-backed with. The E-mail would include an automated form that facilitates the human responses and is easily machine readable. Eventually, the sortie manager's intelligent scheduler would determine what it thought was the best sortie schedule and send it back to the original requester (in charge of the 25 WDs) for confirmation or continued negotiation. This form of negotiation and interaction is what SHAI has already provided in our existing intelligent scheduling systems.

One of the most important functionalities that the intelligent scheduler (and collaborator) can provide is rescheduling in response to dynamic changes. If an instructor canceled at the last minute, the training manager could request ITMS to find an alternate which would cause the fewest perturbations in the training plan. ITMS can do this since it has access to everyone's schedule and knows each person's whereabouts. In the more extreme case, if a fighter unit cancels, ITMS can automatically determine alternatives and contact them. The manager can interact with ITMS to select the best alternatives.

At any time in the scheduling process, humans can intervene and edit the current schedule or define additional constraints or resources. Inside the schedule editor, an explanation as to why the particular resources and time windows were selected is given. Alternative acceptable times can also be displayed. If the user alters the schedule, the intelligent scheduler will check for violations of constraints or over-commitment of resources (including trainees). Once the schedule is finalized and approved, it is automatically published and disseminated to all applicable parties. Since the intelligent scheduler knows all of the resources required for each event, it is a simple matter to send notifications to the manager of each resource. These can be formatted plots showing graphically, for each resource under the manager's control, when each is committed and for which training events.

Inference Engine

The inference engine infers new information and knowledge and makes decisions. In the diagram above, an arrow directed toward the inference engine implies that it uses that information to make an inference and an arrow directed away from the inference engine indicates that it derives and outputs the indicated information. Arrows in both directions indicate both an input and an output relationship. The inferences take several forms. The mastery of skills by each student must be inferred by the ITMS and placed in the student model. This inference for skills for which data directly exists is more straight-forward. The data can come from many sources including course results, course-independent tests, supervisor evaluations, and follow-on course results. Furthermore, since there are relationships between skills in the skill hierarchy, mastery can be inferred for other skills based on mastery estimates for adjacent ones. For example, the ITMS can infer mastery of all subtasks if the supertask is mastered (and vice versa as in an earlier example). Similarly if a more general skill is mastered, all of the more specific skills underneath it can be considered mastered. Additionally, if the student has shown that he can quickly master a number of sibling skills under a more general skill, it may be safe to assume mastery of the more general skill as well.

These previously described types of inferences are one form based on graphical information. Another is based on the career map. The inferencing engine can use the graphical prerequisite links in the career map to assemble the chronological order of events that the student must accomplish to achieve his goals, given his current accomplishments. The inference engine also examines the skill prerequisites of the goals and subgoals, compares them to the student's current levels or the values expected after taking one of the courses, and determines if additional courses are required to address any skill deficiency.

To evaluate a course's ability to meet its learning objectives, the engine uses a combination of constraint satisfaction and statistical inference and uses data from several sources. Consider a very simple example where are there 4 students and 3 courses. Each course has several learning objectives, several skills, it is trying to teach. After taking the course, each student will have several opportunities to have his relevant skills evaluated. This situation is depicted below. Student A has taken Course 1 and 2 as denoted by the arrows. Course 1 happened to have developed a particular skill, Si. At the end of the course, student A will be evaluated in reference to skill Si and this can be used as input to the process of determining the course's ability to teach Si. This is shown by the "SA, C1, Si Results" box. This box is attached to an arrow that is an extension of the arrow from student A to course 1, indicating that evaluation of skills taught to Student A by course 1 will continue to be evaluated over time. The second box up the evaluation arrow indicates data on student A's skill Si mastery from a job performance review by his supervisor. Finally, the last box shows the results of student A's performance at the beginning of a new course, C11, that requires Si as a prerequisite skill. The Decay box indicates that before taking the course, a six-month delay occurs during which time skill Si decayed some amount. Keep in mind that this structure would be replicated for every skill taught by course 1 to student A. And of course this structure is replicated for every student taking each course.

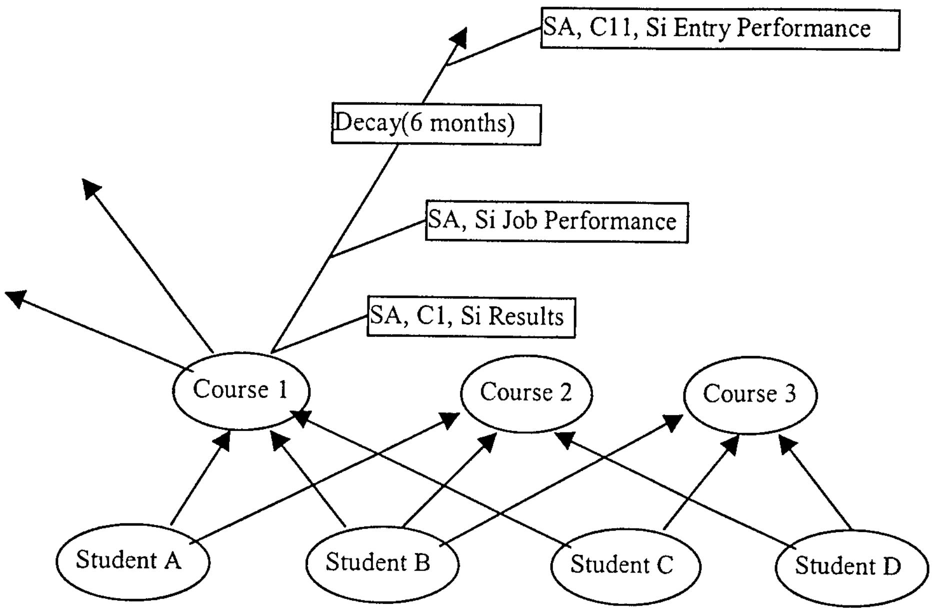


Figure 3. Constraint Satisfaction Network

Constraint satisfaction is used in the following way. The assumption exists that for each skill, a course has some quality in its ability to teach that skill (either specific or general skills). This quality will vary for different types of students. Furthermore, each student has his own unique ability to learn new skills. This can be estimated for a particular student by seeing how well he learns new skills in all of the courses (and other learning opportunities) he experiences. His mastery of skills is given, in a noisy way, by the various evaluations performed on him. Furthermore, if significant time elapses without practicing a skill between when it was learned and when it was evaluated, decay will be assumed to occur. There will be an apriori estimate of this decay for each skill, but it is assumed to vary somewhat for different students. The problem becomes how to most consistently label the graph to explain the various (noisy) evaluation results. This is both a statistical and a constraint satisfaction problem. Any course or student cannot be considered in isolation, as the figure above shows. For example, if the results for course 1 are poor, it may be caused by a poor course, poor students, or a mismatch between the specifics of the course and the attributes of the students. If the students have done well in other courses then the second hypothesis is eliminated. Unless the students are very similar to each other, the third hypothesis cannot be the sole explanation either. Thus, to determine which of the three (or which combination of the three) is most appropriate, the data for all students and all courses must be considered simultaneously. Constraint satisfaction techniques were developed for precisely this type of problem.

The inference engine also reasons about time. When it sends an E-mail notification to which it expects a response, it schedules an event in the future to "timeout" if it has not received the response and take the appropriate action. If the response occurs before that scheduled event, the event is cancelled. Similarly, it will use the decay heuristics to determine when a student's skills should be checked, if he is not exercising them. When that day arrives, the student's recent history is examined to determine if in fact, skill decay has likely occurred, and if so, to take the appropriate action. This is how the inference engine achieves proactivity.

As mentioned previously, the inference engine may decide to make proactive notifications using an E-mail system, so an interface to such a system is shown in the design. Additionally, data from an external database system must occasionally be processed and thus, an interface to these is provided as well.

4.3 Functionality

The ITMS is a general capability that can be customized by the users to manage any system of training courses. This occurs by first creating the skill hierarchy then models for the courses and jobs. The process is completed by input of the career map and miscellaneous heuristic knowledge.

Student-Related Functionality

One of the ITMS's most important functionalities is the pro-active notification of students. This typically occurs via E-mail with an acknowledgement expected. ITMS will follow up with additional E-mails and/or regular mail, if required. ITMS will eventually notify the student's supervisor, if it receives no response. If the student happens to log on his web page during this period, ITMS will explicitly request updated E-mail and contact information. Student response can be via E-mail or the Web. Based on the results of user requirements analysis, we may also provide ITMS the ability to make notifications by telephone and receive some types of responses by phone.

The notifications will include telling them when they need to take prerequisites for future courses and telling them that they're falling behind in completing the prerequisites or other career goals. ITMS will proactively notify them that certain skills may have degraded and need to be evaluated and possibly refreshed, that their next assignment requires a different skill set and therefore refresher or additional training or scenarios, and that courses or job requirements have changed and additional training is needed to stay up-to-date.

ITMS will also provide information to the students via their personal ITMS web page including their strengths, weaknesses, and progress. This will be a bar chart that shows their mastery level of relevant skills. These bar charts will be hierarchical and follow the skills hierarchy. Thus, the first bar chart will correspond to the first level, or breakdown, of the student's skills. The student can then click on any particular bar to see that skill's subskills expanded (based on either the subtask or the more-specific relationship) in its own bar chart. Any of those skills can be similarly selected and so on. The student's web page will have a "What's new" section and a "What's new for him" as well which would contain information about new courses, new versions, or new knowledge required for his specific job or based on courses he has taken.

The ITMS web page will also include a career counseling section where the student can view the career map and select career goals and timelines for achieving them. The ITMS will provide advice as to what timelines are reasonable and achievable. The ITMS will determine, from their career goals and already achieved skills, ranks, jobs, and courses, and from the career map, what subgoals are required to meet the student's objectives and in what order. This will be based on the explicit prerequisite relationships as well as required prerequisite skills. If any of the student's skills meet the required prerequisite levels, the ITMS will find appropriate courses that can build the skill level from what the student possesses to that required for some objective.

The ITMS will proactively question the students (and the ITMS will expect answers via E-mail or web page). It will get feedback on each course they've taken as to its ability to build mastery in the specific and general skills as well as the prerequisite skill levels required. It will get feedback on their current job, what it entails and their ability to meet it. It will give tests and evaluations, if the system suspects skill decay, and provide remedial refresher courses, if appropriate. ITMS starts with heuristic apriori decay constants for each skill but learns actual constants based on skill, skill type, and individual soldier.

Explanation Capability

The inference engine, which makes all decisions in ITMS, will record rationale for each of its decisions. These then provide the basis of an explanation facility for students and other users. For example, the student could ask why the ITMS has included a particular course in his career plan and it might respond with the explanation that it was a prerequisite for one of the prerequisites for one of his career goals. He might ask why the ITMS believes certain skills have decayed and the ITMS would respond with a description of what it believes the student's job currently is and what skills are not practiced by that job along with the rate at which it believes those skills decay for that student. A supervisor might ask for an explanation for why the ITMS as estimated the mastery of a certain skill for the student to be a particular value. The ITMS would respond with a description of how it was calculated and where the supporting data come from. Similarly a course author might ask for an explanation for the ITMS's calculation showing the degree to which the course was meeting one of its learning objectives.

Supervisor and Mentor-Related Functionality

The ITMS will proactively question the supervisors (who answer via E-mail or their web page). It will get feedback on soldier and the preparation for his current job provided by courses he has taken. It will get updated job requirements, both general and specific, for the particular position. It will provide the same bar chart type functionality to view the skills of students under their supervision, subject to the authorization of the training manager. Similar functionality will be provided to the student's mentors. These will also have facilities for receiving and answering student questions regarding their job, career, or courses that they are taking.

Course Author-Related Functionality

Course authors will be provided a graphical editor to maintain the skill hierarchies. The editor will show graphically either one or both hierarchical relationships simultaneously, and provide user-friendly point and click methods for adding new skills, and linking them to other skills through one of the two relationships – subtask or more-specific. A similar interface will exist for selecting which skills from these hierarchies are being taught and to what level by courses under the author's control. In addition to the skills taught, the author will also specify the prerequisite skills and degree of mastery required for successful entry into the course. These skills may be specific and concrete or more general and abstract in nature. He will also graphically specify prerequisite relationships between his course and other courses, jobs, or ranks (by editing the career map). Additional course information includes the version, possible scenarios and their attributes, required hardware or software or other constraints.

He will also be able to get evaluations of his course in bar chart format where his estimates of the prerequisite and resulting skills of the course are compared to evaluator's assessments and actual results from students who have taken the course. Their improvement (or lack thereof) of skills, after having taken the course, will be based on supervisor evaluations of students job performance and the course's ability to prepare them, student self and course evaluations, and performance in subsequent courses. The evaluations will consider the student's improvement in both specific and general skills. Data-mining techniques will also be used to try to differentiate course value between different types of students. In these cases, the ITMS will make suggestions for improvement by identifying course areas that are weak (which skills are not being taught as well as expected) and which course areas are weak for different types of students. It will also suggest when additional scenarios might be needed to cover aspects of a course that aren't currently covered or not covered by enough scenarios based on student use and failure patterns. The ITMS will also provide the author with how many students have taken which versions and/or scenarios and which did the best later.

The ITMS will also provide, to a group of course authors, course configuration management capabilities. Configuration management refers to aiding the courseware development process by tracking the different versions of the separate files that make up the courseware, making sure that all the development team is using the most up-to-date versions, and that the final released product has these most up-to-date versions of files. ITMS will maintain a directory of "published" courseware files which represent the currently accepted version of a course. Authors then "checkout" these files to make updates and the ITMS keeps track of who has what file and when it was checked out. The file then becomes read-only for other team members and they are warned, when they view it, that it is currently being revised, when the revision was started, and who is doing the revision. The ITMS would also archive old versions of files. This scheme keeps authors from making parallel changes to the same files, while continuing to let them reference them in their own work. When the revisions are made an approved, the new version of the file is added back to the directory.

Training Manager-Related Functionality

Training managers will be provided a graphical editor (similar to the course authors) to maintain the skill hierarchies and to edit the skill requirements of jobs, as well as the skills developed or practiced by the jobs or expected on-the-job training. They will also have similar graphical capabilities to edit the career map, though their focus will tend to be on the jobs as opposed to the courses. The ITMS will proactively determine if there are skills required for jobs for which no course or other learning event develops the required skills to the required degree of mastery and notify the training manager.

The ITMS will provide the training manager with an overall view of the students, courses and jobs in his specialty. It will provide him graphically with how many students are currently at each point in the career map. It can also project these numbers into the future, based on how filled each spot is and the time required for the average student to achieve various milestones as well as capacity constraints which might limit throughput. The ITMS can also accept waiver rules from the training manager and show how this affects the particular course in terms of eligible students and expected graduations (based on percentages who expect to pass, given the waivers) or how it affects the overall system, into the future. The ITMS will also provide how many students have taken each course, version, and scenario.

4.4 Innovations

The ITMS includes several innovations:

- ITMS is intelligent. It makes decisions. It "remembers," not just stores, information and knowledge (in the sense that it keeps information in working memory and reacts when certain events do or do not occur). It is proactive. All aspects of the system are stored in an explicit knowledge representation which allows end-users to edit and modify all aspects of the system, subject to the appropriate authorizations, of course.

- ITMS addresses the time span that encompasses an entire career.

- ITMS acknowledges the concept that students, courses, and jobs change over time, so that the corresponding Student Models, Complex Course Models, and Job Models must change as well, even while the history and content of the old versions is preserved.

- The ITMS utilizes user-editable hierarchies of skills, knowledge, and tasks as a basis for the course, job, and student models. Both subtask (to fix a piece of equipment requires the subtasks of trouble-shooting and repairing) and more-specific (the ability to fix a specific radio is a more specific skill compared to the ability to fix any radio) relationships are supported in the multiple inheritance hierarchies.

- ITMS performs several kinds of inferencing. It makes inferencing based on graphical descriptions (prerequisite links and hierarchical relationships), using Constraint Satisfaction, and based on statistics.

- ITMS makes proactive decisions and actions, including proactive E-mail notification.

- Training requirements, resources, events, trainees, instructors, and other ITMS objects, are not just data, but actual intelligent entities which facilitate many different uses. Just as they schedule their corresponding real world object themselves, they could also be made to provide explanations of their actions for training or optimal resource acquisition purposes.

- No one has previously applied advanced scheduling techniques to the training management problem before.

- The concept of distributed collaboration of a mix of automated and human decision makers in separate organizations and locations is innovative. Although applied to a few problems, it has certainly not been applied to training management systems.

5.0 Existing Training Management Systems

Little or no research has been performed for training management system and no system has employed any Artificial Intelligence techniques. Therefore the related work consists primarily of the many training management software systems that have been developed and marketed. These systems are called by various names including training management systems or software, education management systems, computer-managed instruction (CMI), or training administration systems. There are currently over 60 such systems being marketed [Hall 1998]. Companies marketing products include Allen Communication, Asymetrix, American Training International, CBT Systems, Cytation Corporation, DK Systems, Geometrix, Informania, Integrity Training, ITC Learning Corporation, KnowledgeSoft, Lasso Communications, Inc., Learncom, Inc., Macromedia, NETg, On Tour Multimedia, Oracle, Pathlore, Plateau, Saba Software, Inc., Saratoga Group, Silton-Bookman Systems, Inc., Syscom, Inc., Teamscape, TTG Systems, Inc., Micromedium, and Infotec. The most popular products being marketed include Pathware, Librarian, Manager's Edge, Ingenium, Registrar, TrainingServer, AdminSTAR, SkillVantage Manager, PHOENIX, and World Trak.

These systems do not begin to meet the complex needs discussed here and do not contain intelligent features. These systems are primarily networked database systems and store data relating to course catalogs, class schedules, enrollment, student information, transcripts, class evaluations, homework, self-assessments, course authoring, content management, grades/test scores, and rudimentary skills. The primary benefit they provide is that of a pre-customized DBMS with existing interfaces defined to the vendor's own courseware offerings or authoring tools. The primary disadvantages are that they do not attempt to track higher level skills and they do not exhibit intelligence, decision making, or proactively, leaving these functions to the training managers or the students themselves.

6.0 Future Work

6.1 Phase II

The ultimate goal of the Phase II effort is to aid the training managers. The final system will reduce their work load, improve the utilization of scarce resources, and reduce training management lapses. The primary Phase II objective is to develop a full-scale, operational version of ITMS in Phase II. By working closely with Air Force, other DOD units, and commercial training managers and performing an analysis of the requirements of other commercial potential clients, our implementation effort can be directed most appropriately and therefore most efficiently for commercialization. Since the ITMS will be implemented in three major releases, the Air Force and other users will have the opportunity to use it operationally early in the project and provide us the necessary feedback to perfect it during Phase II. In order to allow operational use, we will need to interface ITMS to several existing database systems as described in Section 3.3, Task Descriptions.

6.2 Potential Applications

The primary Phase II project results will be a full-scale, operational Intelligent Training Management System (ITMS) developed in cooperation with several different users. The ITMS will have immediate use throughout the DOD and Federal government. In fact, it has significant support from current DOD training managers. For example, several officers at Tinker AFB in Oklahoma have expressed a strong desire for an ITMS and discussions with them had a strong influence on the Phase II design. They will be some of the users of the Phase II system, *during Phase II.* The US Army's Distance Learning Center at Fort Huachuca, Arizona, has also expressed much interest in the ITMS and plans to use the Phase II system, *during Phase II.* The individuals charged with managing the training process for the Navy are the ships' executive officers (XOs). Commander Pinto, the XO on the USS Paul Hamilton (DDG-60, an AEGIS Destroyer) has stated that training management is one of his primary problems. He has even begun the process (coincidentally) of requesting that the Navy upgrade its training management software, which is not currently at an acceptable level of capability. Thus during Phase II we will have operational users throughout the DOD to make sure the resulting system is beneficial to the government.

Several other great opportunities to marketing the ITMS to the government exist. These primarily relate to the fact that SHAI is one of the premier Intelligent Tutoring System (ITS) developers, and thus has a large base of customers who are interested in training management. For example, our Tactical Action Officer (TAO) ITS, currently in operational use by the Surface Warfare Officers School (SWOS) and onboard the Paul Hamilton, was recently selected by the Navy for use onboard all AEGIS ships. These six dozen ships all have the same training management problem described by Commander Pinto, and since they will all already be using one SHAI product, it will be straight-forward to introduce ITMS to those same ships.

Similarly Paul Losiewicz, of the Air Force Research Laboratory is involved in Air Force intelligence training. He is extremely interested in both our ITS authoring tool and this ITMS project. Furthermore, one of our committed Phase II users is the US Army's military intelligence training group at Fort Huachuca. They already train many Air Force intelligence specialists and have a good relationship with Goodfellow AFB, the primary intelligence training center for the Air Force. Goodfellow AFB also trains many Army intelligence specialists. With this kind of cross training relationship, the ITMS can be expected to quickly migrate from one center to the other.

SHAI is currently developing an ITS authoring tool for use by NASA training managers to create ITSs to teach astronauts the skills to operate in-space experiments. These training managers have the same management problems that ITMS will address.

There are several potential commercial applications. The commercial corporate training industry is currently $62.5 billion for companies with over 100 employees [Training Magazine 1999]. Even only allocating 3% to the management function leads to $1.9 billion in training management costs. An ITMS can greatly reduce much of these costs, indicating that a substantial market exists. This is further validated by the large number of training management systems currently marketed.

Many large corporations, especially those involved with possibly life threatening activities, also have complex training requirements and would benefit by ITMS. Examples include nuclear power, airlines, toxic waste handling and clean-up, chemical factories and oil refineries. Any organization with complex training requirements would benefit from ITMS. Accordingly, Esteem Software Incorporated, our highly successful commercialization partner for other endeavors, has agreed to market the ITMS to their substantial customer.

SHAI has identified the commercial training industry as its primary marketing target and written our business plan around this assumption. Accordingly we have hired a Director of Business Development, Rick Row, whose resume given below in Section 6.0, to pursue it, full-time. He has begun to establish relationships with many of the vendors of training systems and services including CBT (largest vendor of training system to teach software operation), Wicat (largest commercial vendor of aircraft simulations for pilot and maintenance training), Flight Safety (largest commercial provider of aviation training services), Raytheon Training, and NETg. Furthermore he has already identified some 60 training management systems currently being marketed. Since the ITMS encompasses technology significantly beyond all of them, each is a potential partner for licensing our technology to provide it to their clients, through integration with their products. A few success stories during the Phase II will ease the process of approaching these companies. Furthermore, the next 18 months should see a significant shakeout of these dozens of vendors so that it will be clearer with whom we should license our ITMS technology. The Phase II proposal, explicitly includes the task, "User Requirements Definition," which itself includes steps to investigate commercial requirements and existing tools, partly with an eye toward future partnering arrangements. While at EPRI, Mr. Row arranged intellectual property licenses with manufacturers for EPRI technology resulting in $500,000 annual revenue. We expect him to make similar arrangements for the ITMS with vendors of training and training management software and services.

Meanwhile, as partly described above, SHAI has sold millions in intelligent tutoring system products and services. We will approach these government and commercial customers, all of whom also have training management system requirements. By involving them in the Phase II Knowledge Engineering, User Requirements Definition, Design, Installation and Training, and User Evaluation and Feedback tasks, we will ensure that the final Phase II ITMS is both commercially viable and useful for DOD training managers. Since we will have three ITMS releases in Phase II, there will be ample opportunity to incorporate their feedback. Since we will have operational DOD and commercial users, during Phase II, we will have several success stories which we can use to approach our other ITS clients, training management system vendors, and training system and service providers.

Our commercialization strategy has many facets. From our previous experience, we know that commercialization activities cannot wait for the Phase II project to end. SHAI has developed an SBIR commercialization process that begins with the Phase I presolicitations. We identify which topics have the most commercialization potential for SHAI and then pursue those aggressively. This topic offered great potential because it represents the intersection of two of SHAI's strong areas, which have heretofore been completely separate; intelligent scheduling and training. This project effectively leverages off our successes in these two fields and will allow us to approach our training customers with another product and/or service, as described above.

Concurrent with Phase II, we will perform market research in support of the Phase II task, User Requirements Definition. That task includes defining functionality which will make the ITMS more commercially viable. The market research will provide focus for the set of commercial users who are most likely to buy an ITMS. The requirements of these users can be folded in with those of our currently committed users. Concurrent with Phase II will be the development of features required by the commercial marketplace and development of contacts to sell the resulting ITMS.

There are thousands of organizations which could benefit from ITMS. There also appears to be little competition. Many training management systems exist, but these are not automatic, merely logging and keeping track of a user's decisions, primarily regarding attendance and scheduling. Most of these don't even do deconflicting. Thus, our ITMS will have the significant benefit over competitors of being largely automatic and will also be more tailored to complex training requirements.

There are three different business plans as a result of this project. The first is to sell the ITMS itself. The design will be flexible enough that users will be able to define their own kinds of positions (jobs), teams, trainees, skills and knowledge, tasks, training requirements, training events, resources, etc. Once a user has entered this knowledge, the ITMS will be able to automatically track students' skills, proactively notify him of new courses, prerequisites, or if he's falling behind; determine requirements; schedule training events (including needed resources); and track results for individuals or teams. Industries with complex training can be approached directly with ITMS, or we could sell it through companies which currently provide training to them (such as Wicat and Flight Safety, who both serve the aviation industry). These would also leverage off our existing extensive Intelligent Tutoring system marketing efforts.

SHAI has completed intelligent scheduling system projects for NASA and is starting others. SHAI scheduling products are already in use by several organizations and we are expanding our market for such tools. We anticipate that this effort will result in additional scheduling algorithms that we will be able to incorporate into our existing scheduling products, thus increasing the benefits they provide and their value.

Finally, ITMS can be used as a basis to create customized ITMS solutions for individual commercial organizations. Because it is designed for low cost application to new training management problems, we can customize it at a low cost.

7.0 References

[Agrawal et al., 1993] "Mining association rules between sets of items in large databases, "In Proc. of the ACM SIGMOD Conference on Management of Data, 1993.

[Allen, 1984] Allen, J., F., "Toward a General Theory of Action and Time", Artificial Intelligence, 23, 1984.

[Beck et. al., 1997] Beck, J., Davenport, A., Sitarski, E., Fox, M., "Beyond Contention: Extending Texture-Based Scheduling Heuristics," AAAI-98 Proceedings, AAAI Press, Menlo Park, CA, 1997.

[Charniak, 1991] "Bayesian Networks without Tears," AI Magazine, 1991.

[Cheeseman et al., 1988] "Autoclass: A Bayesian Classification System," 5th International Conference on Machine Learning, pp. 54, 1988.

[Fox, 1983] Fox, M. S., Constraint-Directed Search: A Case Study of Job Shop Scheduling, Ph.D. thesis, Carnegie-Mellon University, 1983.

[Hall, 1998] Hall, B. "Training Management Systems: How to Choose a Program Your Company Can Live With," Brandon Hall Resources, http://www.brandon-hall.com, October 1998.

[Henke, 1996] Henke, A., Artificial Intelligence Techniques for Mission Planning and Scheduling Automation- Final Report dated July 1996, Stottler Henke Associates, Inc.

[Henke & Stottler, 1991] Henke, A.L. and R.H. Stottler. (1991), Automated Manifest Planning: An Artificial Intelligence Approach; Phase I Final Report. Research Report. Stottler Henke Associates, Inc., Belmont, CA, (July).

[Henke & Stottler, 1992] Henke, A.L. and R.H. Stottler. (1992), "Artificial Intelligence Techniques for Scheduling Space Shuttle Processing", Proceedings of the Florida AI Researchers Symposium, Ft. Lauderdale.

[Henke & Stottler, 1995] Henke, A., Stottler, R., Artificial Intelligence Techniques for Flight Activities Planning, Stottler Henke Associates, Inc., 1995.

[Knorr & Ng, 1996] "Finding Aggregate Proximity Relationships and Commonalities in Spatial Data Mining," IEEE Trans. On Knowledge and Data Engineering, 1996.

[Laird and Saul, 1993] "Sequence Extrapolation," Machine Learning Conference Proceedings, pp. 1120-1125, 1993.

[Maher, et. al., 1997] Maher, T., Green, J., Goan, T., Artificial Intelligence Techniques for Flight Activities Planning - Phase II Final Report, May 1997, Stottler Henke Associates, Inc.

[Mannila, 1995] "Discovering Frequent Episodes in Sequence," Knowledge Discovery in Databases Proceedings, 1995.

[McVey et. al., 1997] McVey, C., Atkins, E., Durfee, E., Shin, K., "Development of Iterative Scheduler to Planner Feedback", AAAI-98 Proceedings, AAAI Press, Menlo Park, CA, 1997.

[Morgado & Martins] Morgado, E., Martins, J., "CREW_NS: Scheduling Train Crew in The Netherlands," AAAI-98 Proceedings, AAAI Press, Menlo Park, CA, 1997.

[Shatkay & Zdonik, 1996] "Approximate Queries and Representations for Large Data Sequences," ICDE 1996.

[Training Magazine, 1999] "Industry Report 1999", Oct. 1999.

[Zweben & Eskey, 1989] Zweben, M., Eskey, M., "Constraint Satisfaction with Delayed Evaluation" Eleventh International Joint Conference on Artificial Intelligence, Detroit, Michigan, August, 1986, pg. 875.

[Zweben & Fox, 1994] Zweben, M., Fox, M. S., eds., Intelligent Scheduling, Morgan Kaufmann, 1994.

Appendix A Phase I Prototype Design

Introduction

This document is a design specification for the Phase I part of the ITMS project. The purpose of this part of the project is to implement and demonstrate a proof-of-concept system that will utilize AI techniques to improve the training management process (i.e. increase training efficiency, streamline the course revision process, etc.) This project will be done in Allegro Common Lisp dynamic object oriented system.

A.1 ITMS architecture

The intelligent training management system was originally conceived as a standalone application, to be developed in Lisp under Windows. Further knowledge elicitation revealed that there was a defined need for a Web interface to much of the ITMS functionality, and so the Web interface (implemented as a number of CGI scripts written in Perl) was subsequently added to the existing Lisp application. The current system is schematically shown in figure 1. The Phase I setup is less than ideal for a number of reasons, and a number of enhancements will be made in Phase II that should increase the usability of the current ITMS prototype.

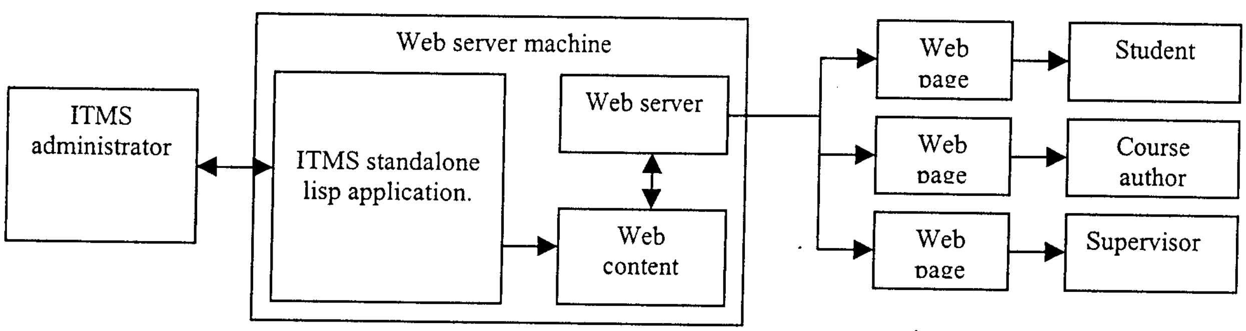


Figure 1.

1.) Single language. Since ITMS consists of two distinct pieces, there is a certain overhead involved in converting data from the form the lisp application understands to the form the web interface understands. This overhead will be eliminated entirely in Phase II, since ITMS will consist exclusively of a Web interface over a database and a reasoning engine.

2.) Perl and Apache integration. In Phase I, the Apache web server was used to generate web content generated by CGI scripts written in Perl. The basic setup used in Phase I was such that the entire perl interpreter, the cgi script, and all the data files used by the script had to be loaded into memory each time the user submitted an HTTP GET or POST request (i.e. each time the user pressed a button on the Web interface). This presented significant overhead, which can largely be eliminated by the use of mod_perl, a perl module that provides Perl/Apache integration. After mod_perl is installed and compiled into Apache, the perl interpreter itself is always resident in memory and doesn't need to be loaded each time, and cgi scripts and data files are cached in memory as well after their first execution. Installation of mod_perl is known to result in two-fold increase in response times.

3.) Script separation and modular design. Additional speedup can be obtained by separating script functionality into distinct scripts (so only a portion of the scripts need to be executed during each GET or POST request).

A.2 ITMS skill graph structure

(not implemented yet)

Predicates (links) relating graph nodes:

Subtask(A, B) True if A is a subtask of B

Example: Subtask(Disassemble power supply, Repair power supply)

PrerequisiteOf(A, B) True if A is a prerequisite of B

Example: PrerequisiteOf(File systems, Networked file systems)

ParentOf(A, B) True if A is a parent of B

Example: ParentOf(Analytic ability, Computer systems)

SkillLevel(A, X) True if the skill level of A is X

Example: SkillLevel(MechanicalRepair, 50)

Inferences we want to make:

$\text{SkillLevel}(B, L) \wedge \text{PrerequisiteOf}(A, B) \rightarrow \text{SkillLevel}(A, L)$
$\forall X\ (\text{Subtask}(X, A) \wedge \text{SkillLevel}(X, L)) \rightarrow \text{SkillLevel}(A, L)$
$\text{Subtask}(X, A) \wedge \text{SkillLevel}(A, L) \rightarrow \text{SkillLevel}(X, L)$
$\text{ParentOf}(A, B) \wedge \text{SkillLevel}(A, L) \rightarrow \text{SkillLevel}(B, C * L),\ 0 < C < 1.$

A.3 ITMS GUI

ITMS GUI will be a conventional Microsoft Windows menu-based GUI. The current GUI structure is shown in figure 2.

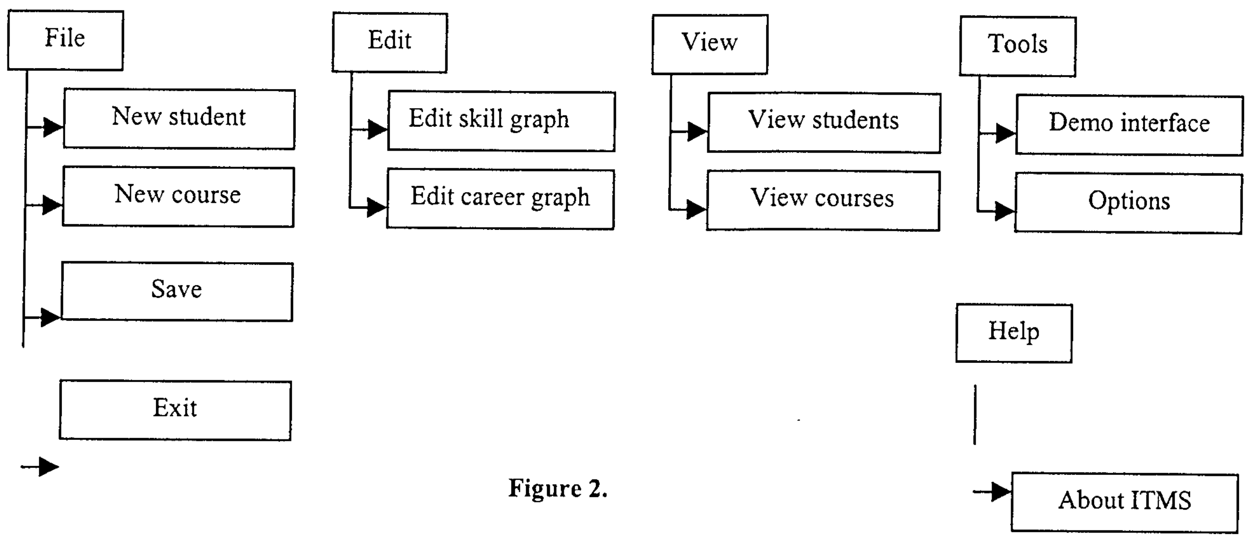


Figure 2.

Figure 3 shows the dialog boxes that make up the ITMS GUI, and how the user navigates through them.

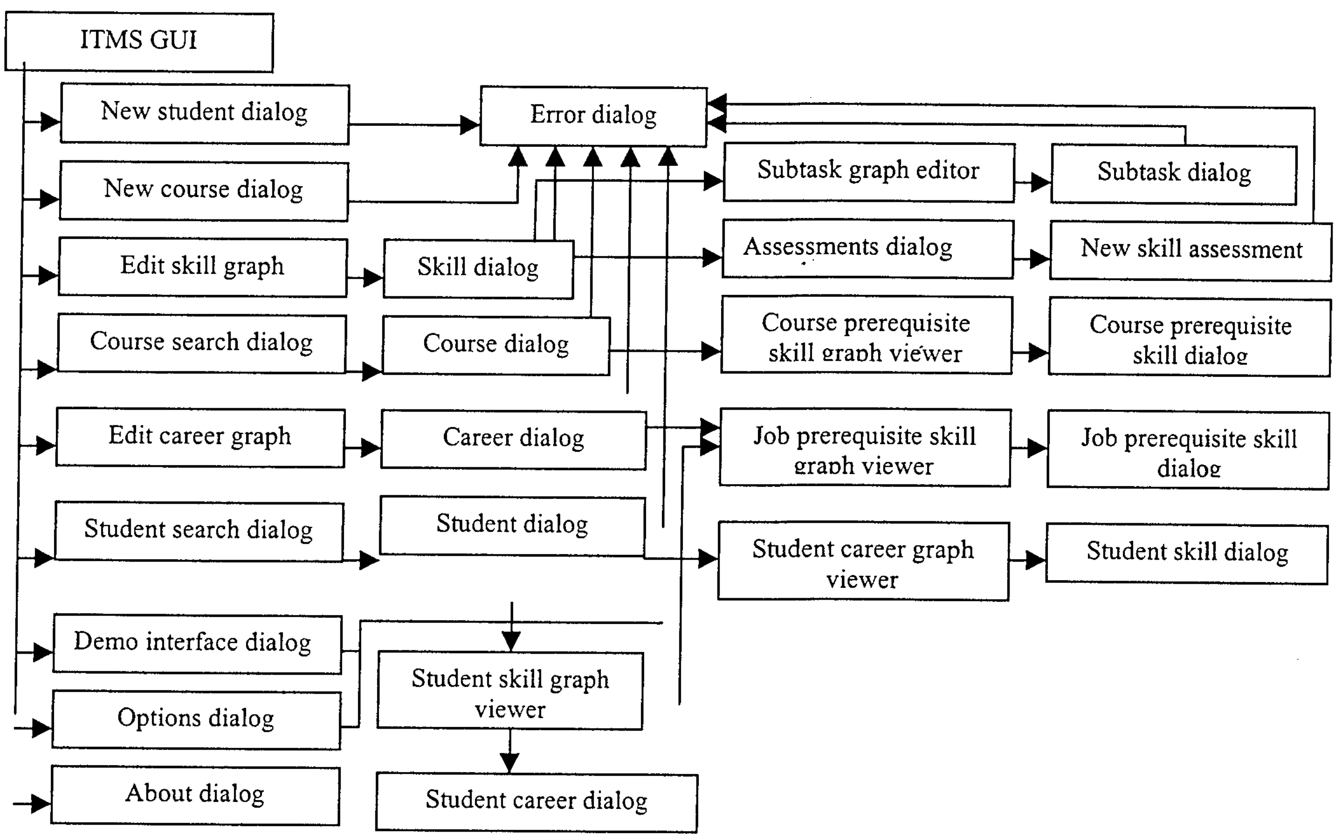


Figure 3.

A.4 ITMS classes and data structures (lisp application)

Class ITMS

;This is the ITMS inference engine.

Slots:

```
:Career-graph       ;the global copy of the career hierarchy (jobs, titles, rank, etc.)
:Skill-graph        ;the global copy of the skill hierarchy

;; These slots can end up consuming large amounts of memory, and so will end up being obtained ;;
from disk as needed (in Phase II at any rate).

:Students           ;student models currently in the system.
:Courses            ;course models currently in the system.
:Schedule           ;a list of events sorted by date.
:Mailbox            ;e-mail interface.
:DL-supervisor      ;contact information for the Distance Learning supervisor.
:Old-versions       ;a list of older versions of career milestones.
:Time               ;current time
```

Methods:

```
;; General methods.
(read-itms       file)            ;reads the ITMS structure from a datafile.
(write-itms      itms    file)    ;writes a given ITMS structure to a datafile.
(init-object     object  stream)  ;initializes ITMS from a given input stream.
(print-object    object  stream)  ;writes ITMS to a given output stream.

(update-skills   graph)           ;propagate global skill hierarchy changes
(update-career   graph)           ;propagate global career graph changes

;; Event methods

(clear-event     id)      ;clears the event associated with id from the event queue.
(trigger-events  date)    ;triggers events scheduled for a particular date.
(schedule        event)   ;Schedules an event for some future date.

;; Scheduling methods     ;These methods schedule various events to occur at a specific time in
                          ;the future.

(schedule-career-survey   student  career-name    days)
(schedule-falling-behind-message   student  days  plan-days)
(schedule-timeout-message          address  cause days)
(schedule-skill-decay     student  days)

;; E-mail creation methods ;These methods create e-mail message that ITMS sends out.

(make-new-course-version-message)
(make-new-career-version-message)
(make-falling-behind-message)
(make-timeout-message)
(make-describe-plan-message)
(make-no-legacy-career-string)
```

(make-create-more-versions-message)
(make-legacy-career-string)
(make-skill-and-course-string)
(make-not-enough-time-message)
(make-goal-accomplished-message)
(make-useful-courses-message)
(make-create-useful-courses-message)
(make-ready-message)

;; E-mail sending methods ;These methods send e-mail messages out.

(send-useful-courses)
(send-ready-message)
(send-course-info-message)
(send-student-info-message)
(send-itms-help-message)

;; E-mail response methods ; These methods are called by mailbox class in response to ; certain e-mails ITMS receives.

(update-skills-and-decay)
(update-career-goal)
(career-completed)
(update-job-skills)
(update-course-skills)
(update-course-enrollment)
(career-assignment)

Class ITMS-gui

;Class defining the main GUI for ITMS.

Slots:

:itms ;the slot holding the itms engine itself.
:new-student-dialog ;Various dialog box classes that ITMS displays in response to user ;action.
:new-course-dialog
:skill-graph-editor
:career-graph-editor
:student-search-dialog
:course-search-dialog
:demo-interface-dialog
:option-dialog
:error-dialog
:about-dialog

Methods:

(initialize-instance :after) ;This method sets a number of variables ITMS-gui depends on.

(new-student) ;These methods are invoked whenever the user selects the appropriate ;action from the menu.

(new-course)
(edit-skill-graph)
(edit-career-graph)
(view-student)

(view-course)
(demo-interface)
(options)
(about-itms)

Class Student

;Class defining the student in ITMS.

Slots:

:first-name	;Personal information...
:last-name	
:e-mail	;The student's e-mail address (very important).
:address	
:city	
:state	
:zip-code	
:supervisor	;The student's supervisor contact information.
:old-skill-levels	;Student's old skill levels (used for skill decay estimation)
:current-list	;The list of current career assignments.
:goals	;The list of career goals.
:plan	;A graph representing the student's plan.
:courses	;A list of courses taken by the student.
:career	;A list of career milestones taken by the student.
:skills	;The student's local copy of the skill graph
:os	;Student's OS
:cpu	;Student's CPU class
:memory	;How much memory the student has.
:disk	;How much disk space the student has.
:cd-rom	;Does the student have a CD-ROM?
:internet	;Does the student have internet access?

Methods:

(update-skills)	;Update student skills to reflect global hierarchy changes
(clear-skills)	;Remove skills no longer present in the global hierarchy
(update-career)	;Update student career to reflect global hierarchy changes
(can-run job)	;Returns true if the student's hardware/software supports a given career ;milestone.
(ready job)	;Returns true is the student has sufficient skills to undertake a given career ;milestone.
(decay-skills)	;Decays the student's skills.

Class Job

;Class representing a career milestone (a course, a job, or a rank, in our domain).

Slots:

:name	;The job name
:number	;Version number
:type	;job, course, or rank.
:supervisor	;Contact info for the supervisor.
:length	;Minimum length one must spend on the job.

:students	;The students previously enrolled here.
:skills	;Skills required and taught by this milestone.
:skills-estimated	;Skill estimates for this milestone.
:os	;OS needed for this milestone (if any).
:cpu	;CPU needed for this milestone (if any).
:memory	;memory needed for this milestone (if any).
:disk	;disk needed for this milestone (if any).
:cd-rom	;Does this milestone require a CD-ROM.
:internet	;Does this milestone require internet access.

Methods:

(update-skills)	;updates skills in response to changes in the global hierarchy.
(clear-skills)	;clears skills no longer present in the career hierarchy.

Class Message

;Class representing an e-mail message.

Slots:

:data	;The message itself.
:attachments	;Any attachments to the main message body.

Methods:

(Send message)	;Sends a message through the e-mail system.

Class Skill

Slots:

:name	;The skill name
:decay	;The default decay constant for the skill
:expertise	;the expertise of a particular student in a skill.
:level-needed	;skill level prerequisite for a particular job or rank.
:level-developed	;skill level developed at a particular job.
:assessments	;a list of ways to assess the skill (sorted by assessment cost).

Methods:

Class Graph

;A non-intrusive implementation of a directed graph container. Supports arbitrary data structures ;as nodes, can apply functions to nodes in hashed, or depth first order (so far). Adds and removes ;nodes efficiently (with a function that cleans up edges that is called after a series of node inserts ;and removals).

Slots:

:roots	;the ids of root nodes of the graph
:acyclic-p	;is true if the graph has no directed cycles
:nodes	;the nodes of the graph
:depth	;the maximum depth of the graph

Methods:

(insert-node node id parent-ids child-ids)	
(remove-node id)	;removes a node and all incoming and outgoing edges.
(remove-subtree id)	;removes a node and its entire subtree.
(remove-node-splice id)	;removes a node, but connects all of its parents to all of its children.
(insert-edge parent-id child-id)	
(remove-edge parent-id child-id)	
(update-graph)	;is called after a series of insert-node or remove-node calls. Inserts and ;removes edges to make the graph consistent.
(apply-to-graph function)	;applies a specified function to all nodes in the graph.
(dfs pre-visit post-visit)	;applies pre-visit and post-visit functions to all nodes in the graph in ;depth first order.

Class Graph-node

;Graph node wrapper around data structures stored in the graph class.

Slots:

:id	;the id of the graph node, used for fast retrieval
:data	;the actual data stored in the node
:parents	;the ids of parents of the node
:children	;the ids of children of the node
:pre-visit	;used for cycle detection and graph traversals in specified order
:post-visit	;used for cycle detection and graph traversals in specified order
:depth	;the depth of the given node

Methods:

None

ITMS classes and data structures (web interface)

The web interface classes and data structures mimic the lisp data structures in Perl. The script lisp_to_perl.pl reads the ITMS data file into perl data structures which are then prompty serialized. Then, whenever the CGI scripts need certain information about the student or a career milestone, the appropriate file is read in. While the lisp application maintains one datafile, the web interface stores information for each student and each course and career milestone in its own file. This is done for efficiency and safety.
We can ultimately expect large numbers of requests for this information, and we don't want to read in more information from file than necessary to satisfy a particular query (which will always be about a particular student, or a particular career milestone). Having to read in an entire datafile for each request would be prohibitively expensive.

ITMS server side authentication

ITMS uses the port of Apache web server for Windows. In order to let only authorized personnel access ITMS information on the web, server side authentication has been implemented. Authorization is granted and removed by means of two scripts, add_person.pl and remove_person.pl, respectively. The syntax is:

```
perl add_person.pl [name] [authorization_group] [password]
perl remove_person.pl [name] [authorization_group]
```

When the first script is ran an additional person is granted access with login [name], and password [password] to all information permitted to [authorization_group]. Some examples of authorization groups are Student, Course_author, Supervisor, Administrator.

Appendix B Phase I Prototype User Guide

The phase I implementation of the ITMS project consists of two heterogeneous parts: the GUI written in Allegro Common Lisp, and the web interface implemented as a series of Perl CGI scripts. The two parts interact through a file interface. Whenever ITMS generates a data file it gets converted to the format the Perl scripts can understand.
The ITMS GUI is meant to be ran periodically on the machine containing the web server and the CGI scripts. The GUI can be used to add and remove students, and edit the skill and career hierarchies. Furthermore, all e-mail communication currently goes through the GUI. The web interface is used to display useful information ITMS has collected and inferred on the web.

B.1 ITMS GUI Guide

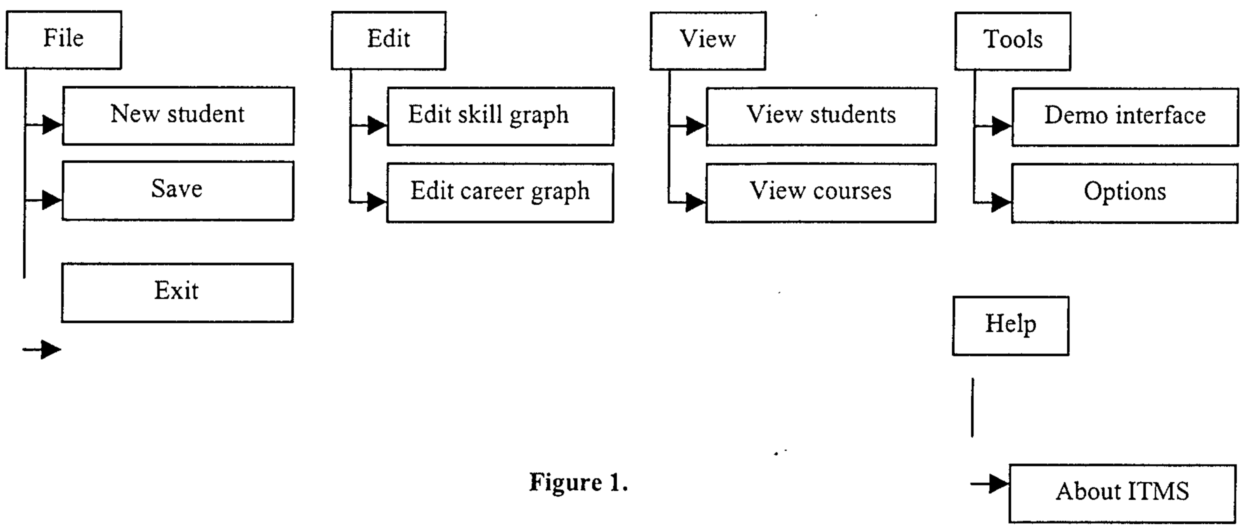


Figure 1.

Figure 1 shows the top-level choices available to a user of the ITMS GUI. Here is a short description of what each item does:

New student: Pops up the new student dialog box that allows the user to enter new student information, specify skills the student may have, etc.

The following buttons are available in the student dialog:

OK: This finalizes the new student's settings and, if the user didn't forget to enter something, adds the student.

View skills:
View career:
Courses taken: These three buttons view student information, however in the case of a new student this information is not yet available.

Cancel: This cancels everything, and doesn't add the student.

Save: Saves the current state of ITMS into a datafile, and converts the file into a format understood by Perl CGI scripts.

Exit: Saves the current state of ITMS and exits from the GUI.

Edit skill graph: Pops up the skill graph editor dialog box that allows the user to modify the global ITMS skill graph (and edit individual skills). No conventional buttons are available in this editor, all actions are performed used the toolbar buttons. There are seven of these buttons and they look like this:

 **New node:** This pops up a new skill dialog. If the user doesn't cancel out of that dialog box, the new skill will be added.

 **Delete node:** This removes the currently selected skill (if any) from the global skill hierarchy.

 **Connect node to child:** One must have already selected a skill prior to clicking this button. Then the next skill you select will become a child of the currently selected skill.

 **Connect node to parent:** One must have already selected a skill prior to clicking this button. Then the next skill you select will become a parent of the currently selected skill.

 **Delete child:** One must have already selected a skill prior to clicking this button. Then the next skill you select will cease to be a child of the currently selected skill.

 **Delete parent:** One must have already selected a skill prior to clicking this button. Then the next skill you select will cease to be a parent of the currently selected skill.

 **Lose focus:** Pressing this button causes the editor to unfocus any skill that was previously selected.

Edit career graph: Pops up the career graph editor dialog box that allows the user to modify the global ITMS career graph (and edit individual career milestones). No conventional buttons are available in this editor, all actions are performed used the toolbar buttons. These buttons are identical to those found in the skill editor. The only changes are that the new node button will pop up a new career milestone dialog box, and all editor changes affect the career hierarchy, not the skill hierarchy.

View students: Pops up the search dialog that allows the user to search for individual students already in the system. This dialog shows the number of students currently in the system, a scroll-box that allows the user to select the criterion to search by (first name, last name, e-mail, etc)., a text box allowing the user to type their search key word, and a series of buttons:

Search: Returns a list of matches to the user's query.

New search: Resets the search, removes all matches.

OK: Commits all changes, and exits from this dialog box.

Delete: Deletes the selected match from the system.

Cancel: Is identical to OK in this context.

Furthermore, double-clicking on any match will bring up a dialog box showing information on the match (in this case a student dialog).

View courses: Pops up the search dialog that allows the user to search for individual courses already in the system. (Note: courses distinct from the career graph are deprecated).

Demo interface: Pops the demo interface dialog box that allows the user to move ITMS time backwards and forwards, and to receive 'e-mails.' The dialog box has three text fields that allow the user to specify the month, day, and year that ITMS considers to be 'today.' Furthermore, there is a text field where a user can enter a file name that will be read in by ITMS as an 'incoming e-mail.' There are also two buttons:

OK: This exits the demo interface dialog, and sets 'today's date' to be whatever the user last set in the date text fields.

Receive message: This 'receives an e-mail message' corresponding to the file specified by the user in the appropriate text field.

Options: Pops up the options dialog box that allows the user to modify e-mail settings (the server address, mail protocols, etc.) Not currently used, since not all of the e-mail functionality is fully in place.

About: Pops up the about dialog box. This box contains ITMS copyright information.

B. 2 ITMS Web Interface Guide

The main ITMS web page contains three buttons:

Login: This buttons triggers Apache server side authentication. In order to proceed, the user must provide a correct login and password. If the user succeeds in doing so, he will obtain credentials corresponding to the group his login is in for the duration of the browser session. If the user successfully logs in, he ends up in a page corresponding to the group his login is in. Let's assume the user is in the 'student' group. Then when he logs in, he will see a page displaying his name, contact information, etc. There will also be 5 buttons:

Refresh: This refreshes the page, committing any changes the user made to his information.

View skills: This takes the user to a page showing his skills in bar chart form. If the skill has children, then clicking on the bar corresponding to the skill will produce a different bar chart showing the sub skills. There are 2 buttons on this page:

Back: This takes the student back one level in his bar chart browsing. If the student as already seeing the bar chart corresponding to all the root nodes in the graph, he is taken to the student's page, but if not, he is taken to a page showing the bar chart containing the parent of the skills he currently sees.

Back to student's page: self-explanatory.

What's new: This takes the user to a page similar to the main 'what's new' page, the only difference being that only changes and updates to career milestones the student already accomplished are shown here.

Career counseling: This takes the user to a career counseling page that allows him to select career goals and get advice from ITMS. The page shows a scroll-list containing all career milestones the student accomplished, a text box where the user can enter the number of days he is allocating for achieving his career goal, and a list of yet-unaccomplished career goals. There are two buttons on this page:

Show plan: This button takes the student to a page showing the plan calculated to achieve his goal. The plan page always has the following two buttons:

Back to career counseling page: self-explanatory

Back to student's page: self-explanatory

Furthermore, if the plan contains career milestones which require some skill improvement on the part of the student, a button labeled 'find courses' appears next to each such career milestone. Then clicking that button will list all career milestones which will improve the student's skills to the required levels.

Back to student's page: self-explanatory.

Back to main page: self-explanatory.

What's new: This button takes the user to a page containing a list of hyperlinks corresponding to new career milestones. If the user attempts to follow a hyperlink here, he will have to get past Apache server side authentication (we do not want any person on the Internet to view career milestone information). The only button here is:

Back to main page: self-explanatory.

Career milestone information: This button takes the user to a page containing a list of hyperlinks corresponding to all career milestones currently in the system. If the user attempts to follow a hyperlink here, he will have to get past Apache server side authentication. The only button here is:

Back to main page: self-explanatory.

Appendix C. Phase I Prototype Demonstration Sequence

C. 1 Installation instructions

Insert the Allegro CL CD into the CD-ROM drive, follow installation instructions.
Insert the ITMS Demo CD into the CD-ROM drive, open a copy of Window Explorer, and click on D:\install.bat. Or, from DOS prompt, type the following two commands:
cd D:
install

When the installation wizard prompts for the directory to install Apache, select c:\sys\apache.
When the installation wizard prompts for the directory to install ActivePerl, select c:\sys\perl.

C.2 Notes

C.3 Webpage outside firewall: http://207.214.202.162/cgi-bin/main.cgi

Local webpage: http://localhost/cgi-bin/main.cgi

Demo reset script: new.bat (located in c:\projects\itms2\web)
Lisp to perl script: convert.bat (located in c:\projects\itms2\web)

Name: Frederick
Password: olo44rin
Name: Jackson
Password: pa55w0rd
Name: Remily
Password: pa55w0rd

(these are all case sensitive)

Demo script

Preparation sequence starts here.

Start apache web server locally. Create a new DOS prompt, and type:
c:\sys\apache\apache –k start
Minimize the DOS prompt window.

Create another new DOS prompt, and make c:\projects\itms2\web the current directory.

Run the demo reset script, new.bat (don't worry about any 'File not found' messages).
(Takes about 20-30 seconds to run)

Start a copy of Windows Explorer, and make c:\projects\itms2\demo the current directory (the output e-mail files will be sent here, and will be called outputfile*.txt, where * is an integer.

Start Allegro CL development environment. Click on File → Open Project. Go to the 'Look in' scrollbox and select c:\projects\itms2, then double click on file 'project 2.'
This will open the ITMS project in Allegro CL.

Preparation sequence ends here.

Demo sequence starts here.

Start a copy of Internet Explorer, and go to local webpage (http://localhost/cgi-bin/main.cgi). Login as Jackson, browse around (nothing should show up under "what's new."). View Jackson's skills, and click on 'stability and support operations.' Only 2 bars should be displayed since Jackson only has proficiency in 2 subskills.

- **Start another copy of Internet Explorer, and go to webpage outside firewall (http://207.214.202.162/cgi-bin/main.cgi). Try to access web page as Frederick, fail, then call in for permissions to be set.**
- **At this point, the permissions czar types the following at the DOS prompt in the directory c:\projects\itms2\web:**

c:\sys\perl\bin\perl add_person.pl Frederick supervisor olo44rin.
(to remove Frederick's permissions, the czar would need to type:
c:\sys\perl\bin\perl remove_person.pl Frederick supervisor).

- **(Once permissions are set either web page can be used by Frederick, and the local web page is recommended since it will likely be faster).**
- **Log in as Frederick, on the remote page, go to career milestone information page, and look at skills taught by 'stability and support refresher DL,' then look at skill required by '96B20 tactical' job. Note that 96B20 tactical now requires a new skill 'predict potential military operations other than war,' a skill not taught by the DL usually taken before 96B20.**

Frederick will now add a new DL course version that teachers the new skill.
Start ITMS by going to the Allegro CL development environment, and clicking Run → Run Project (or alternatively, by clicking on a shortcut that looks like a blue triangle facing right).

Once in ITMS, go to Edit → Career graph editor, and scroll to 'stability and support refresher DL.' Click on 'skills,' scroll to 'predict potential military operations other than war,' and set the skill level developed to 70. Click OK twice, then change the CPU type to '586' (the new course version will have greater hardware requirements), and change the career version to 2. (NOTE: if the version is not changed, ITMS will apply changes to the older version). Click OK. The new DL course version has been added. Furthermore, since Teri Jackson have taken a version of this course in the past, she gets an e-mail notification about a new version. You should now be in the career graph editor. Make 'stability and support refresher DL' a parent of '96B20 tactical,' by clicking the node corresponding to the DL, clicking 'Connect Node to a Child' button, then clicking '96B20 tactical' node.

Click OK to exit the editor, and click File → Save. This causes ITMS to both save its current state to a data file and to run convert.bat to convert the ITMS data file to a form the web interface can understand. Since changes were made to 'stability and support refresher DL,' it appears in the general "what's new" section, as well as in individual "what's new" sections of those students who had taken the course (in our case Teri Jackson).

Teri Jackson receives an e-mail about a new course version, looks in "What's new" section, sees a new course, and sees a new skill that wasn't taught before. She takes the new course (to notify ITMS of this, go to ITMS application, click on Tools → Demo interface, type in 'refresher' in the message file box, and click 'receive message.' This simulates an e-mail notification. ITMS updates Teri Jackson's skill values, and sends a course survey e-mail to Teri. Now type in 'refresher_survey' in the message file box, and click 'receive message.' This simulates Teri's e-mail response to the course survey. ITMS updates the estimated skill values of the new course. Click File → Save to save the changes and convert them to web interface form.

Now when Teri Jackson views her skills, the new skill appears under 'stability and support operations.'

When Frederick views career milestone information for 'stability and support refresher DL', and clicks on skill estimates, he can see that Teri's excellent results on the DL she took are influencing the skill estimates (she got 100 proficiency whereas the original skill levels taught by the course were only 70, thus the estimated values moved up to about 85).

Start another copy of Internet Explorer, and login as Remily.
Go to the career counseling page, and select '96B40 tactical' as a career goal, and click 'Show plan.' The page generated will show the topological ordering of all career milestones Helen Remily will need to accomplish. The page will also show the older versions of courses if the newer versions run on faster hardware than what Helen has. Furthermore, if any career milestones require a skill improvement, the page will generate a list of skills that need to be improved. The user can then select a skill from the list, and click 'find useful courses,' to return a list of courses that raise the skill to required levels.

Now in ITMS, click Tools → Demo Interface, type 'Helengoal' in the message file box, and click 'receive message.' This simulates an e-mail goal request from Helen. ITMS generates an e-mail message that contains much the same information as the web page, and sends the message to Helen.
Now, enter '12' in the month text box, and click OK. This advances simulated time by 60 days. This is enough to trigger a proactive warning from ITMS that warns Helen that she may be falling behind. Now go back to Tools → Demo Interface, enter '2' in the month text box, and '2000' in the year box (ITMS is Y2Kcompliant, by the way), and click OK. This advances simulated time by another 60 days, which is enough to generate a number of proactive notifications to Helen, the last of which is a skill survey prompted by possible skill decay (only skills not used at Helen's current job assignment, 96B10 tactical, are decayed).

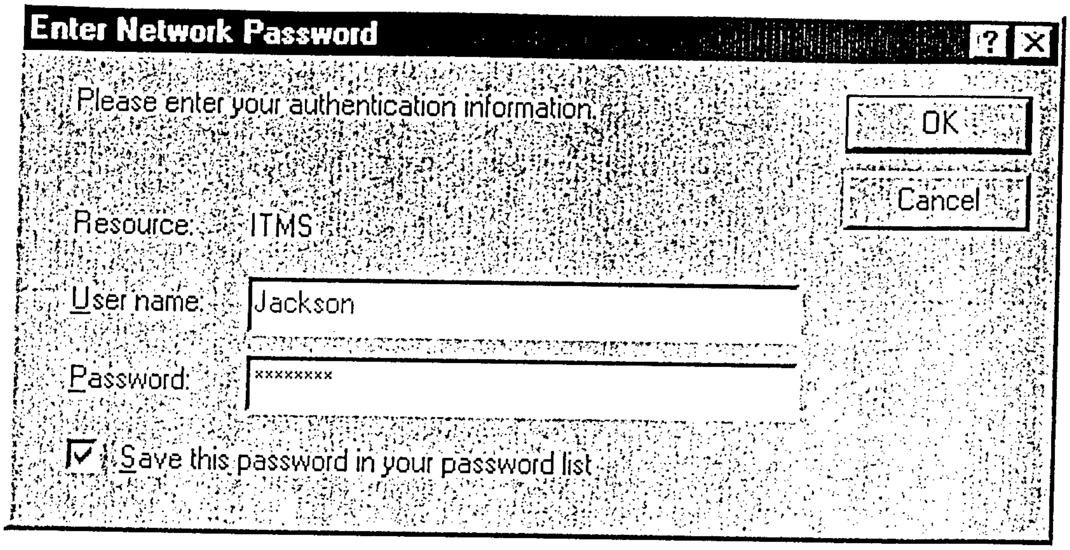
Demo sequence ends here.

Appendix D. Demonstration Sequence (Screen Dumps)

MS-DOS Prompt

8 x 14

```
C:\WINDOWS>cd \projects\itms2\web

C:\Projects\ITMS2\web>new.bat

C:\Projects\ITMS2\web>rem This script initialized the demo sequence for the ITMS
 Phase I.

C:\Projects\ITMS2\web>
        1 file(s) copied
num students = 2
num courses = 0
num skills = 96
num career milestones = 18
num old versions = 0
0 new courses stored.
18 new career milestones stored.
num students = 2
num courses = 0
num skills = 96
num career milestones = 18
num old versions = 0
0 new courses stored.
0 new career milestones stored.
C:\Projects\ITMS2\web>
```


Enter Network Password
?
Please enter your authentication information.
OK
Cancel
Resource: ITMS
User name: Jackson
Password: ********
Save this password in your password list

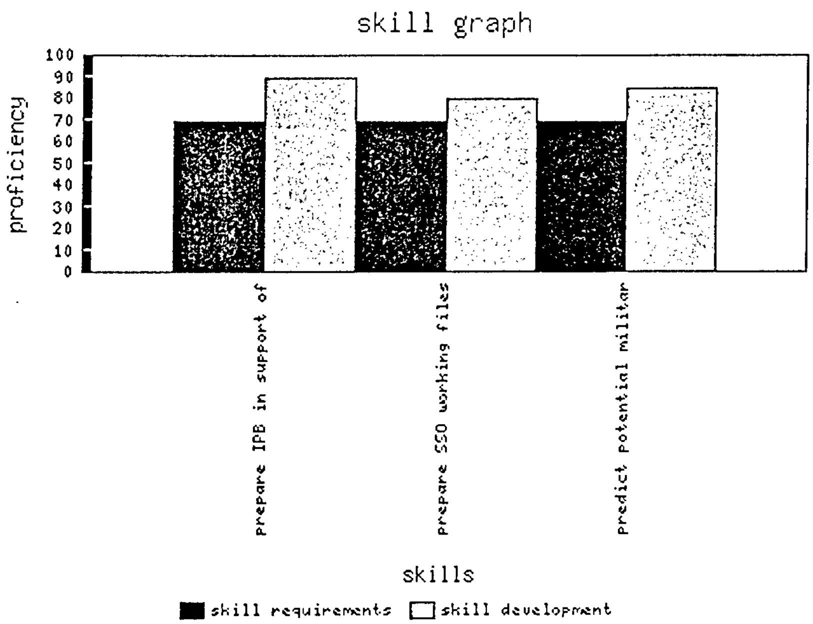
Frederick - Microsoft Internet Explorer

File Edit View Go Favorites Help

Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Print Edit

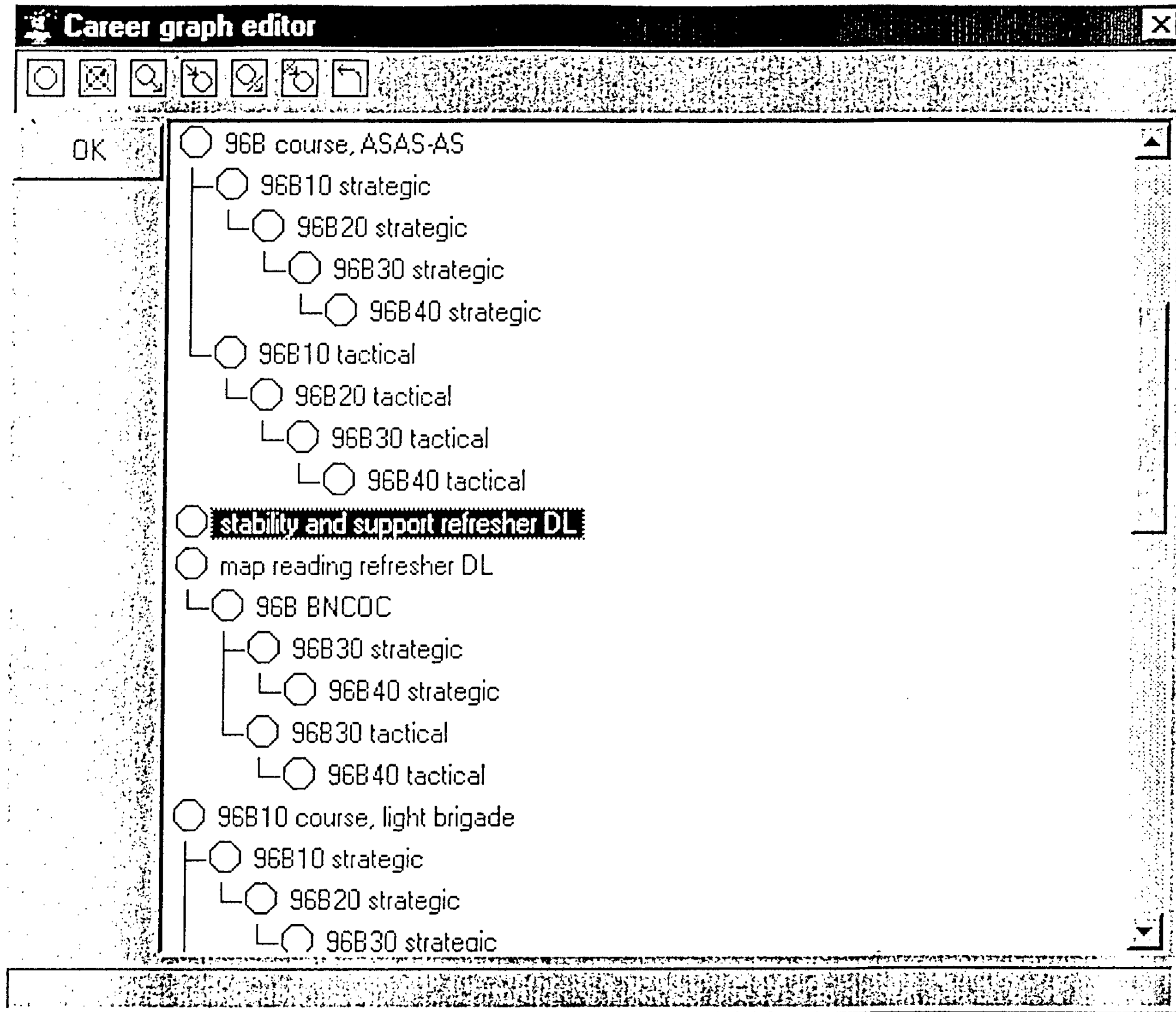
Address http://localhost/cgi-bin/util/view_assigned_skills.cgi?name=96B20+tactical&username=Frederick&version=1 Links

skill chart

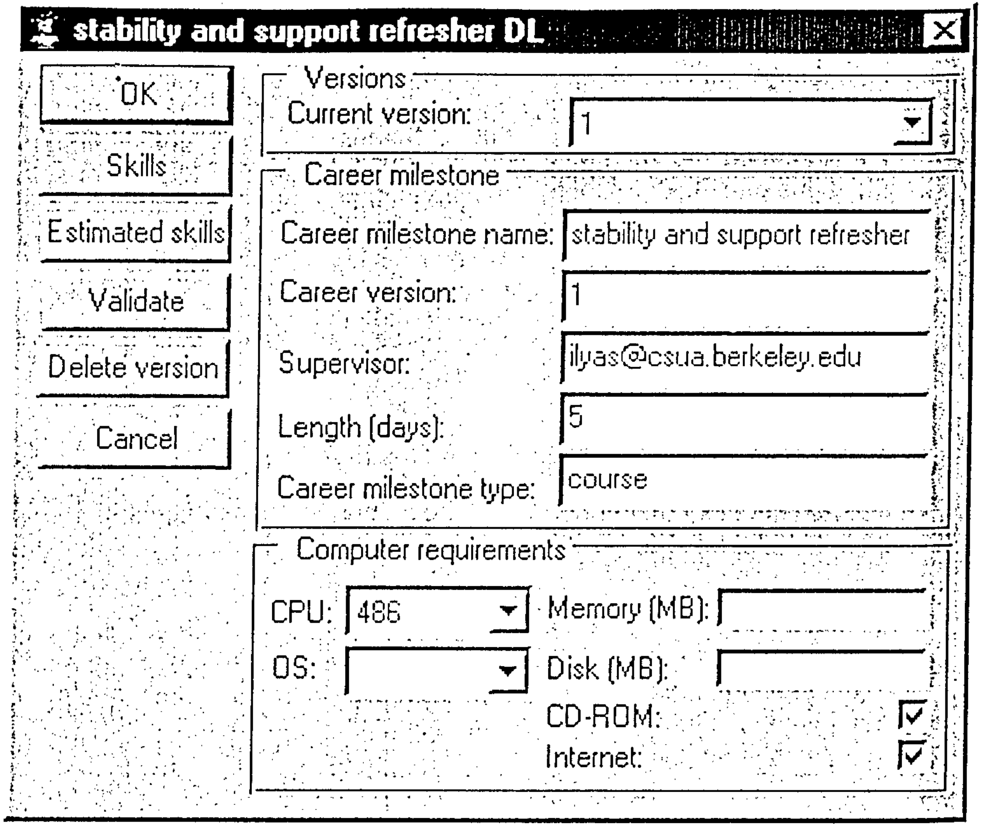


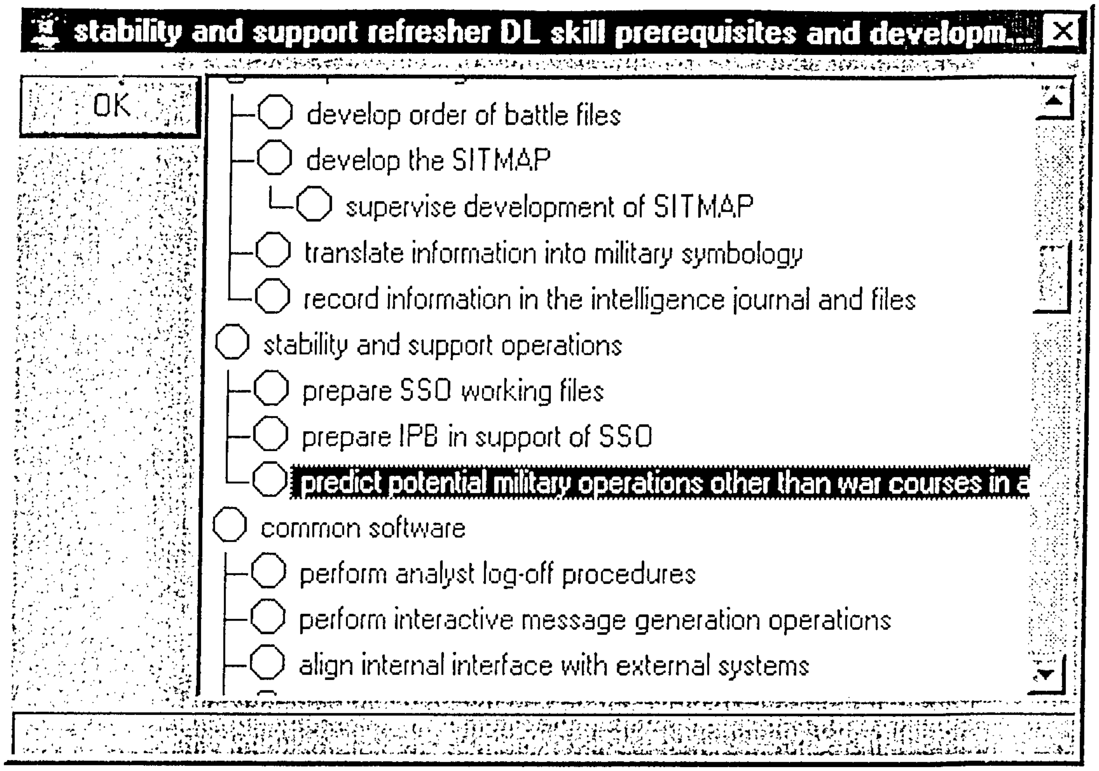
Back | Back to career milestone information page

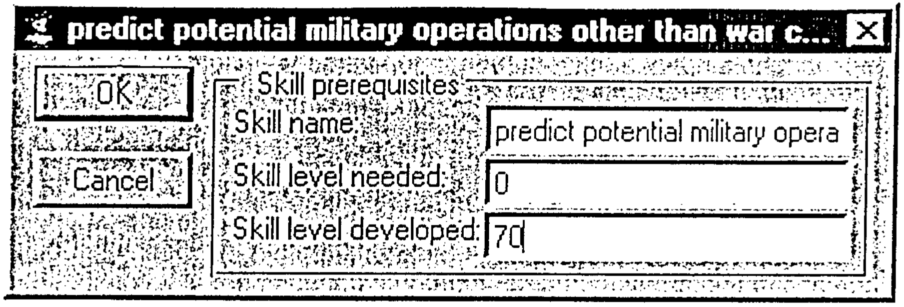
Done | Local intranet zone

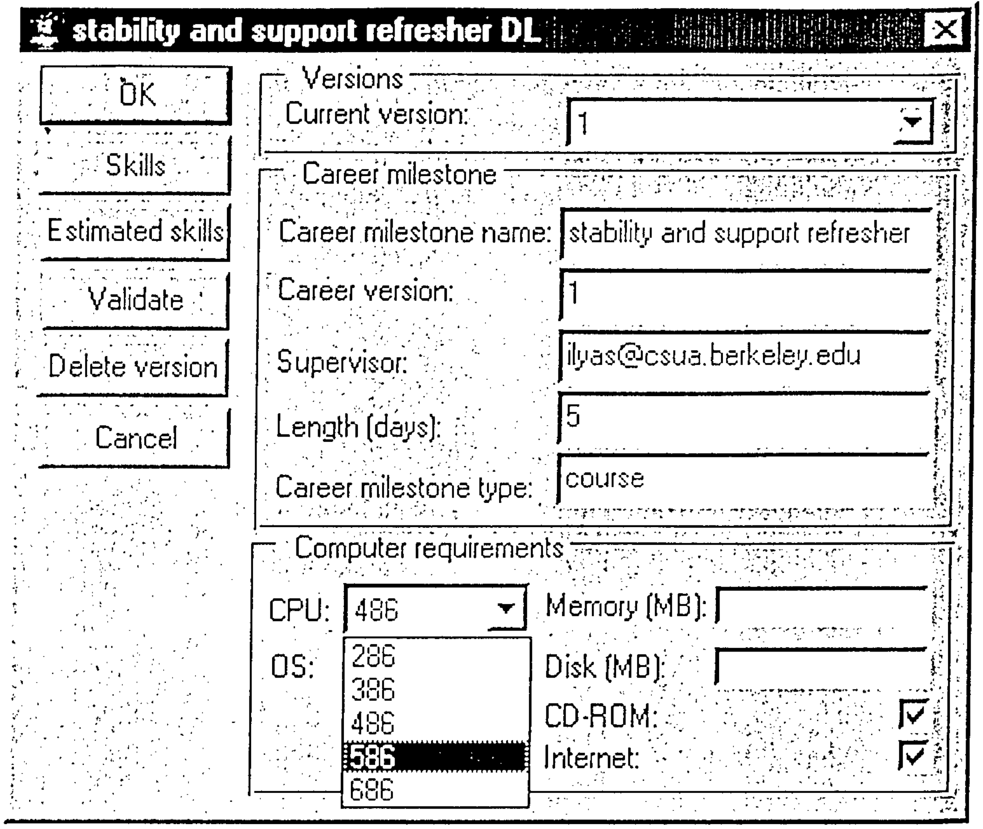


Career graph editor
OK
96B course, ASAS-AS
96B10 strategic
96B20 strategic
96B30 strategic
96B40 strategic
96B10 tactical
96B20 tactical
96B30 tactical
96B40 tactical
stability and support refresher DL
map reading refresher DL
96B BNCOC
96B30 strategic
96B40 strategic
96B30 tactical
96B40 tactical
96B10 course, light brigade
96B10 strategic
96B20 strategic
96B30 strategic


stability and support refresher DL
OK
Skills
Estimated skills
Validate
Delete version
Cancel
Versions
Current version:
1
Career milestone
Career milestone name:
stability and support refresher
Career version:
1
Supervisor:
ilyas@csua.berkeley.edu
Length (days):
5
Career milestone type:
course
Computer requirements
CPU:
486
Memory (MB):
OS:
Disk (MB):
CD-ROM:
Internet:


stability and support refresher DL skill prerequisites and developm...
OK
develop order of battle files
develop the SITMAP
supervise development of SITMAP
translate information into military symbology
record information in the intelligence journal and files
stability and support operations
prepare SSO working files
prepare IPB in support of SSO
predict potential military operations other than war courses in a
common software
perform analyst log-off procedures
perform interactive message generation operations
align internal interface with external systems

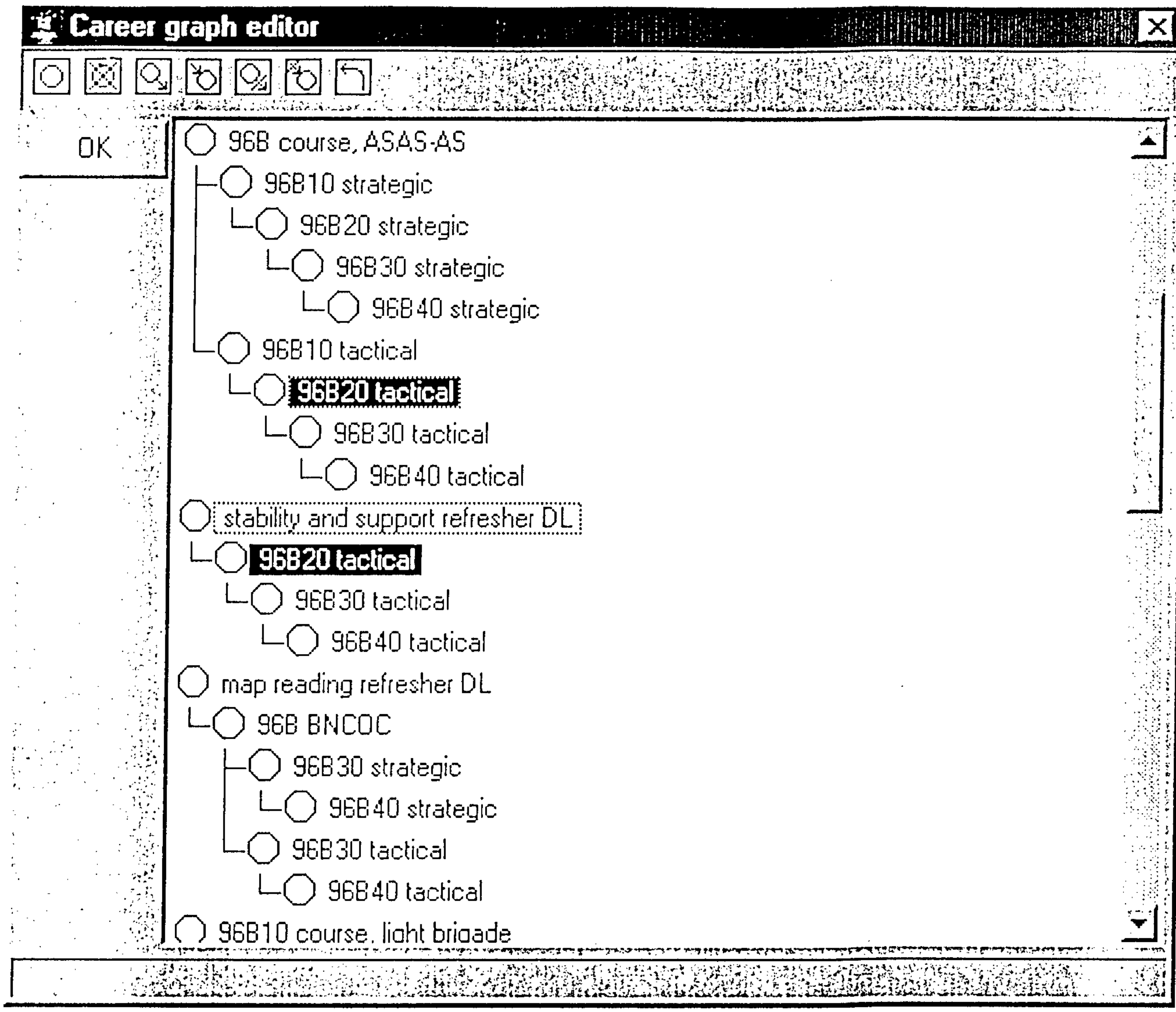

predict potential military operations other than war c...
OK
Cancel
Skill prerequisites
Skill name:
predict potential military opera
Skill level needed:
0
Skill level developed:
70


stability and support refresher DL
OK
Skills
Estimated skills
Validate
Delete version
Cancel
Versions
Current version:
1
Career milestone
Career milestone name:
stability and support refresher
Career version:
1
Supervisor:
ilyas@csua.berkeley.edu
Length (days):
5
Career milestone type:
course
Computer requirements
CPU:
486
OS:
286
386
486
586
686
Memory (MB):
Disk (MB):
CD-ROM:
Internet:

outputfile1 - Notepad

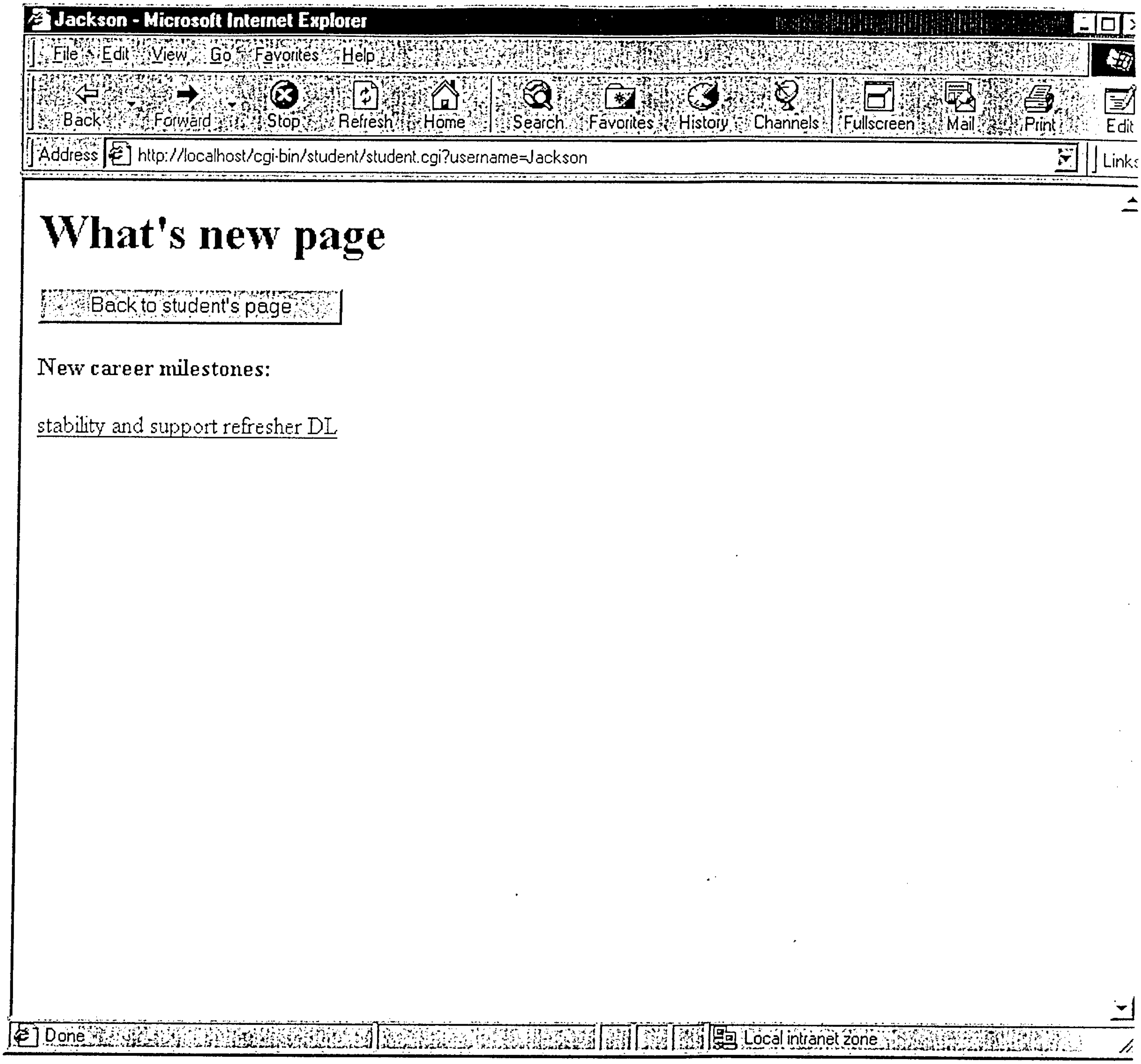
File Edit Search Help

```
Subject: new career version: stability and support refresher DL
This is to inform you that the career milestone you have achieved,
stability and support refresher DL, has changed.  The new version number is 2.


Career milestone: stability and support refresher DL
  type:    course
  length:  5 days
  version: 2

Skill levels needed and developed:

Skill: stability and support operations
  level needed:    0
  level developed: 70

Skill: prepare SSO working files
  level needed:    0
  level developed: 70

Skill: prepare IPB in support of SSO
  level needed:    0
  level developed: 70

Skill: predict potential military operations other than war courses in action
  level needed:    0
  level developed: 70


1 students enrolled in the past:

Jackson
```


Career graph editor
OK
96B course, ASAS-AS
96B10 strategic
96B20 strategic
96B30 strategic
96B40 strategic
96B10 tactical
96B20 tactical
96B30 tactical
96B40 tactical
stability and support refresher DL
96B20 tactical
96B30 tactical
96B40 tactical
map reading refresher DL
96B BNCOC
96B30 strategic
96B40 strategic
96B30 tactical
96B40 tactical
96B10 course, light brigade

PERL

```
C:\Projects\ITMS2\demo>rem This batch file runs the lisp to perl convertion scri
pt, and is callable

C:\Projects\ITMS2\demo>rem from the lisp application using <excl:run-shell-comma
nd>.

C:\Projects\ITMS2\demo>
num students = 2
```


Jackson - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Print Edit
Address http://localhost/cgi-bin/student/student.cgi?username=Jackson
Links
What's new page
Back to student's page
New career milestones:
stability and support refresher DL
Done
Local intranet zone

Jackson - Microsoft Internet Explorer

File Edit View Go Favorites Help

Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Print Edit

Address http://localhost/cgi-bin/student/student.cgi?username=Jackson Links

This is the student's page

First name	Teri
Last name	Jackson
E-mail	teri.jackson@brooks.af.mil
Address	2504 Gillingham Dr., Bldg 170, Suite 25
City	Brooks AFB
State	TX
Zip code	78235
Supervisor e-mail	teri.jackson@brooks.af.mil
OS	
CPU	586
Memory (MB)	
Disk (MB)	
CD-ROM	no
Internet	no

Refresh | View skills | What's new | Career counseling | Back to main page

Done Local intranet zone

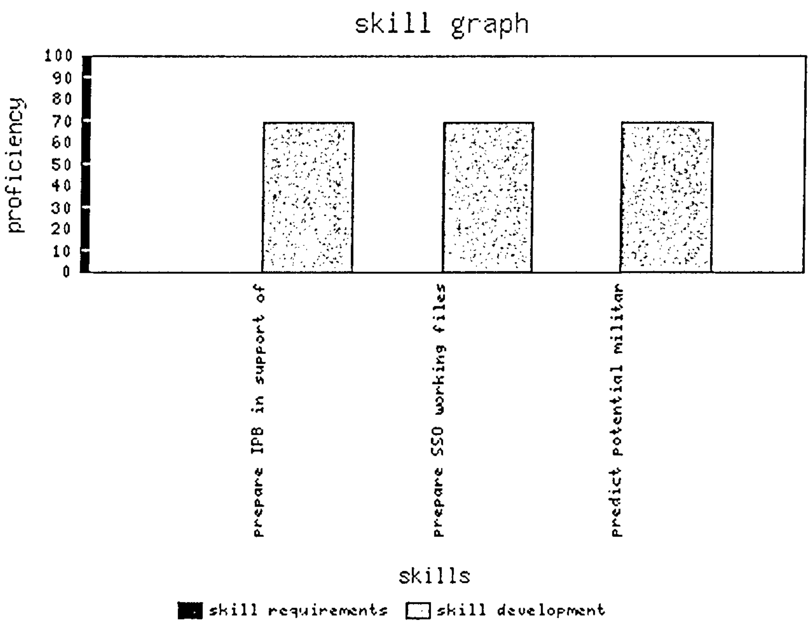
Jackson - Microsoft Internet Explorer

File Edit View Go Favorites Help

Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Print Edit

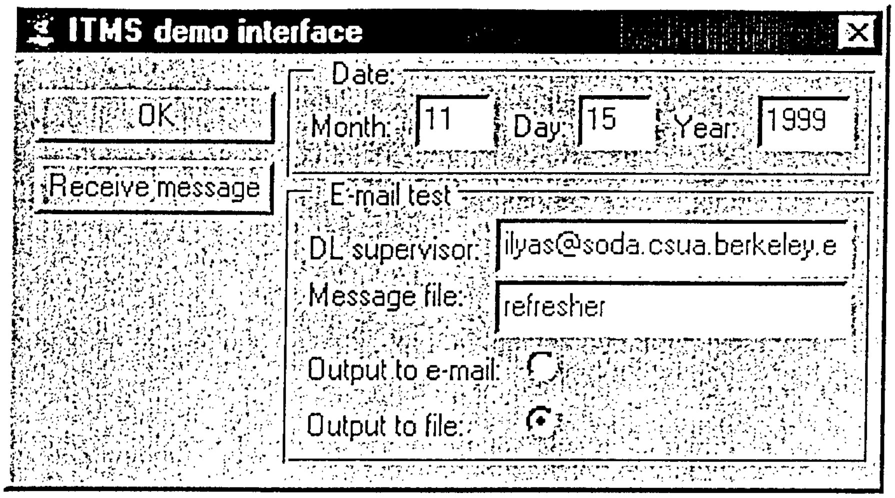
Address http://localhost/cgi-bin/student/view_skills.cgi?name=stability+and+support+refresher+DL&username=Jackson&version=2 Links

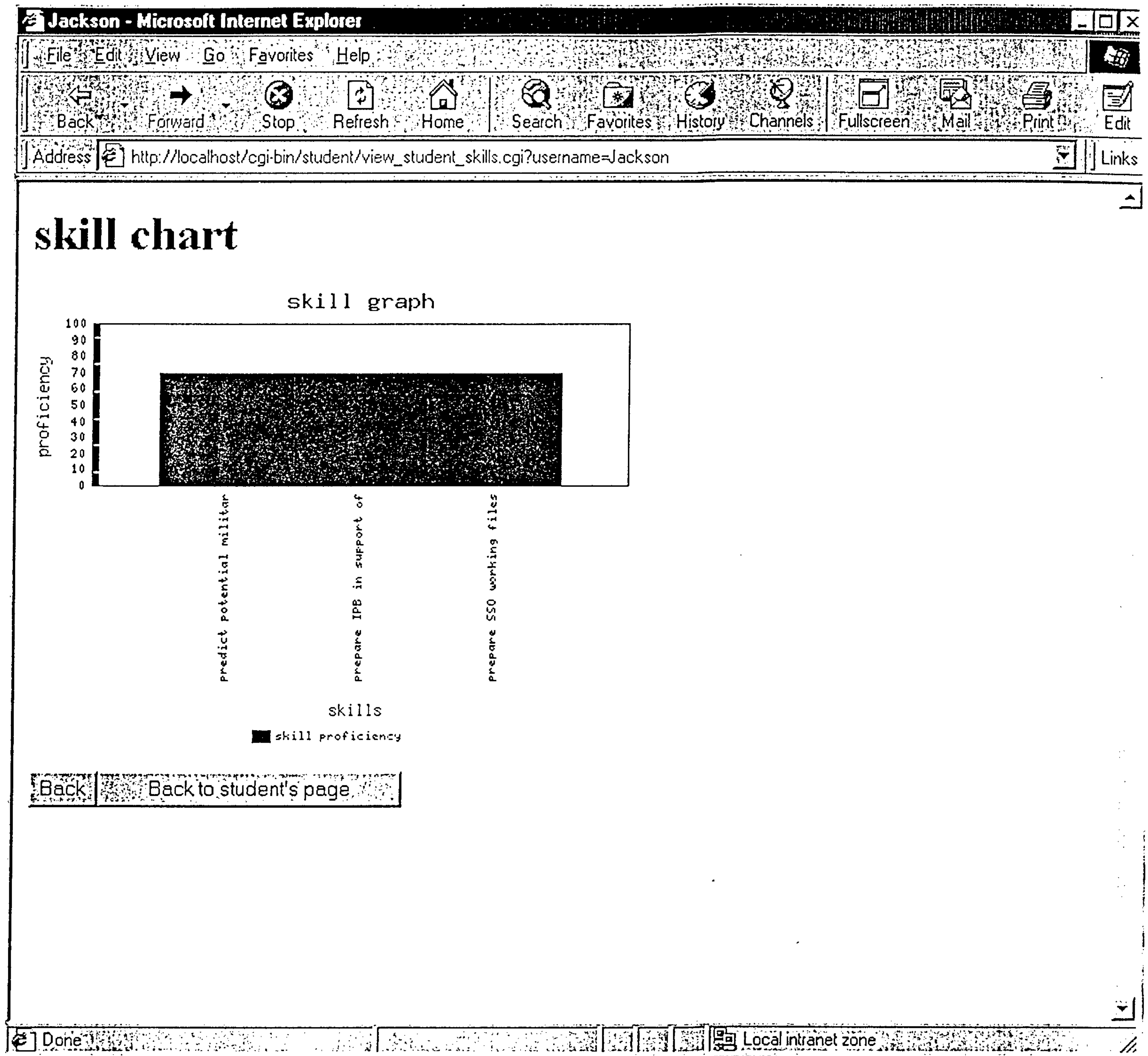
skill chart



Back | Back to what's new page

Done | Local intranet zone


ITMS demo interface
OK
Receive message
Date:
Month: 11
Day: 15
Year: 1999
E-mail test
DL supervisor: ilyas@soda.csua.berkeley.e
Message file: refresher
Output to e-mail:
Output to file:


Jackson - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Print Edit
Address http://localhost/cgi-bin/student/view_student_skills.cgi?username=Jackson
Links
skill chart
skill graph
proficiency
100
90
80
70
60
50
40
30
20
10
0
predict potential militar
prepare IPB in support of
prepare SSO working files
skills
skill proficiency
Back
Back to student's page
Done
Local intranet zone

Jackson - Microsoft Internet Explorer

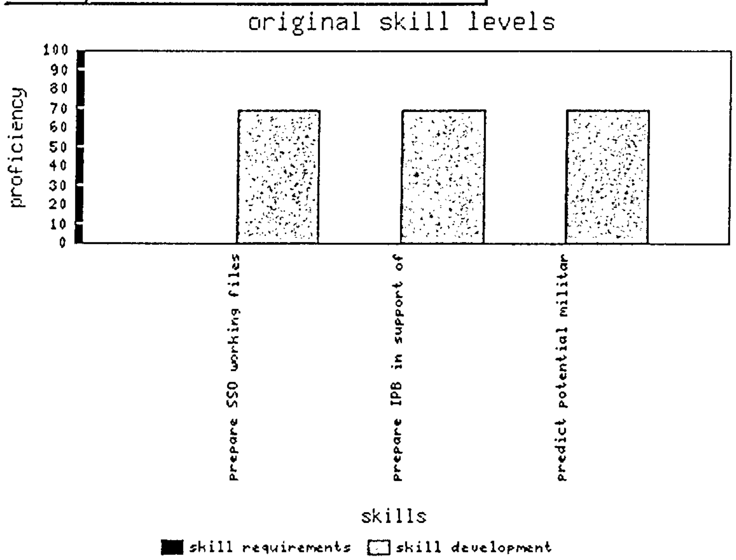
File Edit View Go Favorites Help

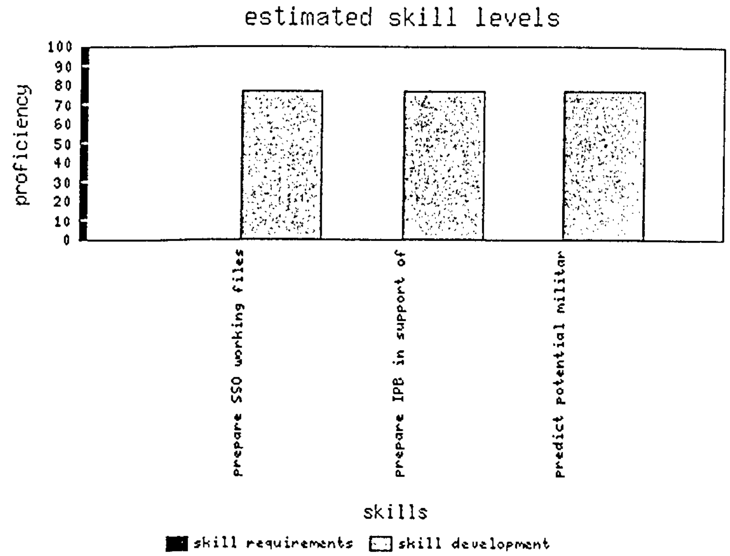
Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Print Edit

Address http://localhost/cgi-bin/util/view_estimated_skills.cgi?name=stability+and+support+refresher+DL&username=Jackson&version=2 Links

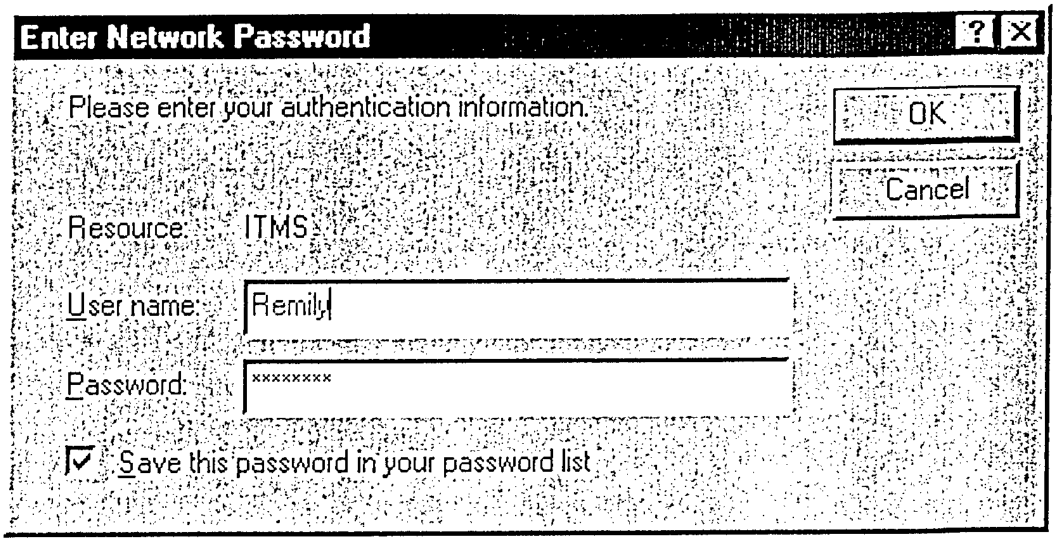
Original and estimated skill levels

Back | Back to what's new page





Done Local intranet zone


Enter Network Password
Please enter your authentication information.
OK
Cancel
Resource: ITMS
User name: Remily
Password: ********
Save this password in your password list

Remily - Microsoft Internet Explorer

File Edit View Go Favorites Help

Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Print Edit

Address http://localhost/cgi-bin/student/student.cgi?username=Remily Links

Please choose your career goal

Show plan | Back to student's page

Career goals accomplished:

96B10 course, basic skills

Time to reach the career goal: 10 days.

Choose career goal:

96B10 course, basic skills

96B10 course, heavy brigade
96B10 course, collection management
96B10 tactical
96B10 strategic
stability and support refresher DL
96B20 tactical
96B30 tactical
96B20 strategic
96B30 strategic
96B40 tactical
96B40 strategic

Done Local intranet zone

Remily - Microsoft Internet Explorer

File Edit View Go Favorites Help

Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Print Edit

Address http://localhost/cgi-bin/student/student.cgi?username=Remily Links

For your information, the following career milestones must be reached before the career goal

- map reading refresher DL
- 96B ANCOC
- 96B BNCOC
- 96B10 tactical
- stability and support refresher DL
- 96B20 tactical
- 96B30 tactical
- 96B40 tactical

Of these, you are currently eligible for

- map reading refresher DL
- 96B10 tactical
- stability and support refresher DL

Furthermore, you will become eligible for the following if you improve your skills

Name	Skills to improve	
96B ANCOC	map reading	find useful courses

The following career milestones do not have versions that will run on your computer

map reading refresher Dl

Done Local intranet zone

Remily - Microsoft Internet Explorer

File Edit View Go Favorites Help

Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Print Edit

Address http://localhost/cgi-bin/student/student.cgi?username=Remily Links

Career counseling page

Back to career counseling page | Back to student's page

Career goal:

96B40 tactical

Insufficient time to reach career goal: 10 days, (minimum time to reach career goal is 140 days.)

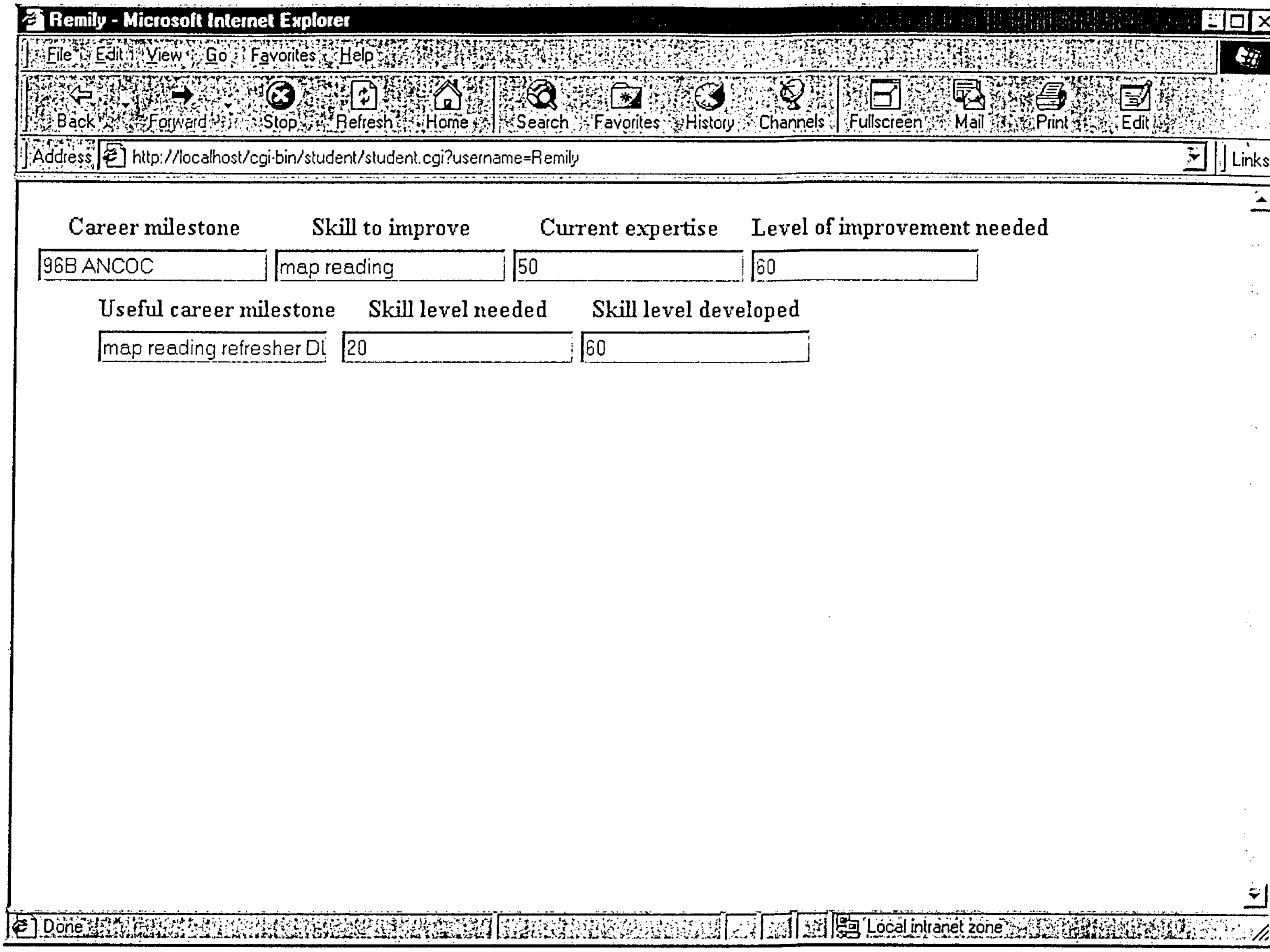
For your information, the following career milestones must be reached before the career goal

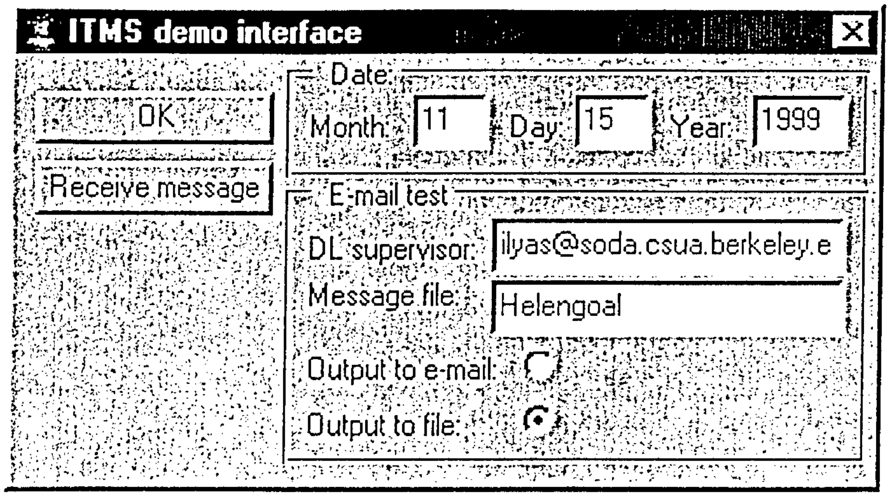
map reading refresher DL
96B ANCOC
96B BNCOC
96B10 tactical
stability and support refresher DL
96B20 tactical
96B30 tactical
96B40 tactical

Of these, you are currently eligible for

map reading refresher DL
96B10 tactical

Done Local intranet zone


Remily - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Print Edit
Address http://localhost/cgi-bin/student/student.cgi?username=Remily
Links
Career milestone
Skill to improve
Current expertise
Level of improvement needed
96B ANCOC
map reading
50
60
Useful career milestone
Skill level needed
Skill level developed
map reading refresher DL
20
60
Done
Local intranet zone


ITMS demo interface
OK
Receive message
Date
Month 11
Day 15
Year 1999
E-mail test
DL supervisor: ilyas@soda.csua.berkeley.e
Message file: Helengoal
Output to e-mail:
Output to file:

outputfile3 - Notepad

File Edit Search Help

```
Subject: Not enough time to reach career goal.

ITMS detected that the time alloted for student Helen Remily
to reach career objective 96B40 tactical (10.0 days) is not sufficient.
The minimum amount of time to reach the career goal is 140 days.
Either the,student hasn't notified ITMS of courses already taken,
or career milestones already accomplished, or the time allowed for
reaching the career objective must be increased.

For your information, the following career objectives must be reached
before the career goal:

stability and support refresher DL
96B10 tactical
96B BNCOC
96B20 tactical
map reading refresher DL
96B ANCOC
96B30 tactical
96B40 tactical

Of those career objectives, you are currently eligible for:

stability and support refresher DL
96B10 tactical
map reading refresher DL

Furthermore, you will become eligible for the following if you improve
your skills:

career milestone: 96B ANCOC
  skill needed: map reading at level 60
    useful career milestone: map reading refresher DL develops map reading to 60


The following career milestones do not have versions which
can run on your computer:

map reading refresher DL

Persons responsible for maintaining these
career milestones have been notified, and versions for your computer
configuration will soon be available.
```

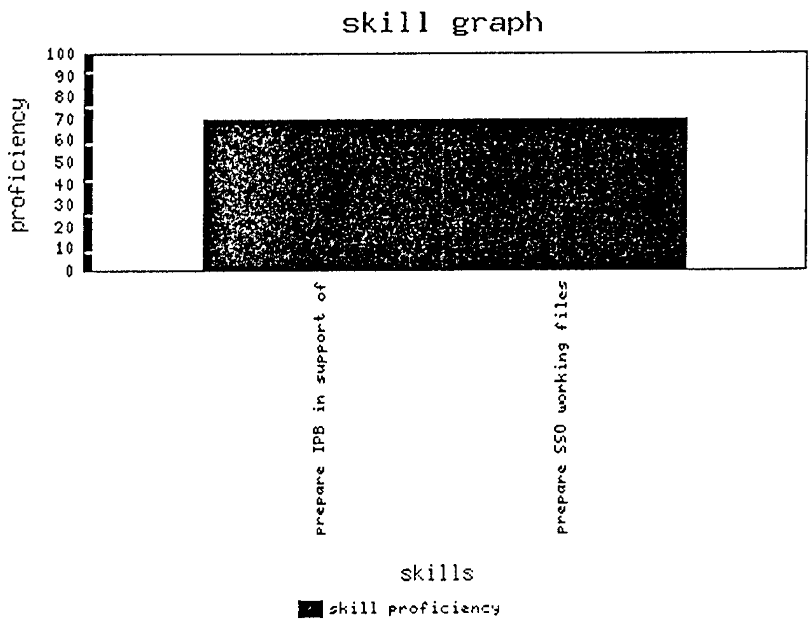
Jackson - Microsoft Internet Explorer

File Edit View Go Favorites Help

Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Print Edit

Address http://localhost/cgi-bin/student/view_student_skills.cgi?username=Jackson Links

skill chart



Back | Back to student's page

Done

Local intranet zone

outputfile4 - Notepad

File Edit Search Help

```
Subject: Helen Remily falling behind.

ITMS has detected that the student Helen Remily only has 5.0 days
to reach his/her career objective 96B40 tactical.  Please make sure the student
fulfills all the necessary prerequisites so the career objective will
 be reached in a timely manner.  For your information it will take at least
140 to reach the career objective.

Thank you.
```

outputfile8 - Notepad

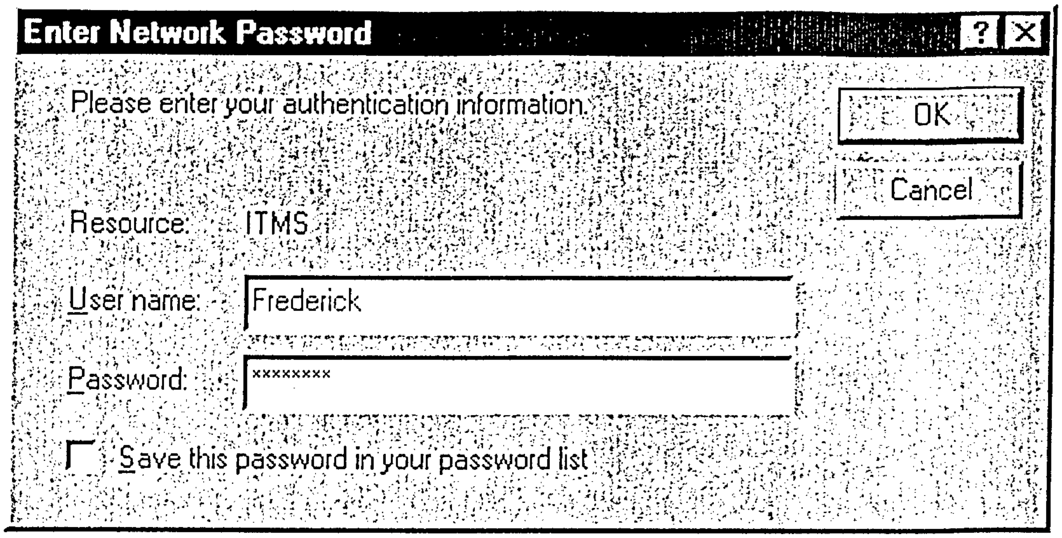
File Edit Search Help

```
Subject: ITMS student survey Remily

ITMS has detected that certain skills of student
Helen Remily
may have decayed over the last 100 days.
Please assess the skill levels of this student
and e-mail the results back.
Thank you.

--
prepare IPB in support of SSO
?
?
prepare SSO working files
?
?
stability and support operations
?
?
create situation alarm criteria
?
?
situation development
?
?
database management
?
?
maintain dynamic databases
?
?
```

outputfile9 - Notepad

File Edit Search Help

```
Subject: teri.jackson@brooks.af.mil not replying to mail.

The following address: teri.jackson@brooks.af.mil
has not replied to ITMS mail for 100 days.
Please make sure the address is still valid.

Thank you.
```


Enter Network Password
Please enter your authentication information.
OK
Cancel
Resource: ITMS
User name: Frederick
Password: ********
Save this password in your password list

401 Authorization Required - Microsoft Internet Explorer

File Edit View Go Favorites Help

Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Print

Address http://207.214.202.162/cgi-bin/login/login.cgi Links

Authorization Required

This server could not verify that you are authorized to access the document requested. Either you supplied the wrong credentials (e.g., bad password), or your browser doesn't understand how to supply the credentials required.

Apache/1.3.9 Server at http://207.214.202.162 Port 80

Done Internet zone

Untitled Document - Microsoft Internet Explorer

File Edit View Go Favorites Help

Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Print

Address http://localhost/cgi-bin/main.cgi Links

Career milestone information

Back to main page

96B10 course, basic skills

map reading refresher DL

96B ANCOC

96B10 course, light brigade

96B BNCOC

96B course, ASAS-AS

96B10 tactical

96B10 course, collection management

96B10 strategic

96B20 tactical

96B30 tactical

96B20 strategic

Done Local intranet zone

Frederick - Microsoft Internet Explorer

File Edit View Go Favorites Help

Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Print Edit

Address http://localhost/cgi-bin/util/career_node_info.cgi?name=stability+and+support+refresher+DL Links

Title	stability and support refresher DL
Version	1
Type	course
Duration	5
Supervisor	ilyas@csua.berkeley.edu
CPU	486
Memory (MB)	
Disk (MB)	
CD-ROM	no
Internet	no

Refresh | Skills | Skill estimates | Back to career milestone information page

Done Local intranet zone

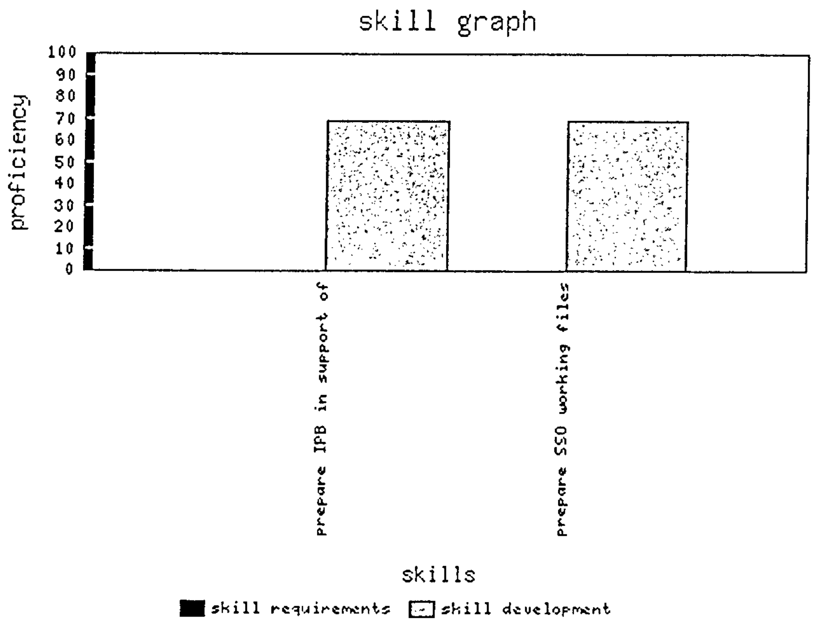
Frederick - Microsoft Internet Explorer

File Edit View Go Favorites Help

Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Print Edit

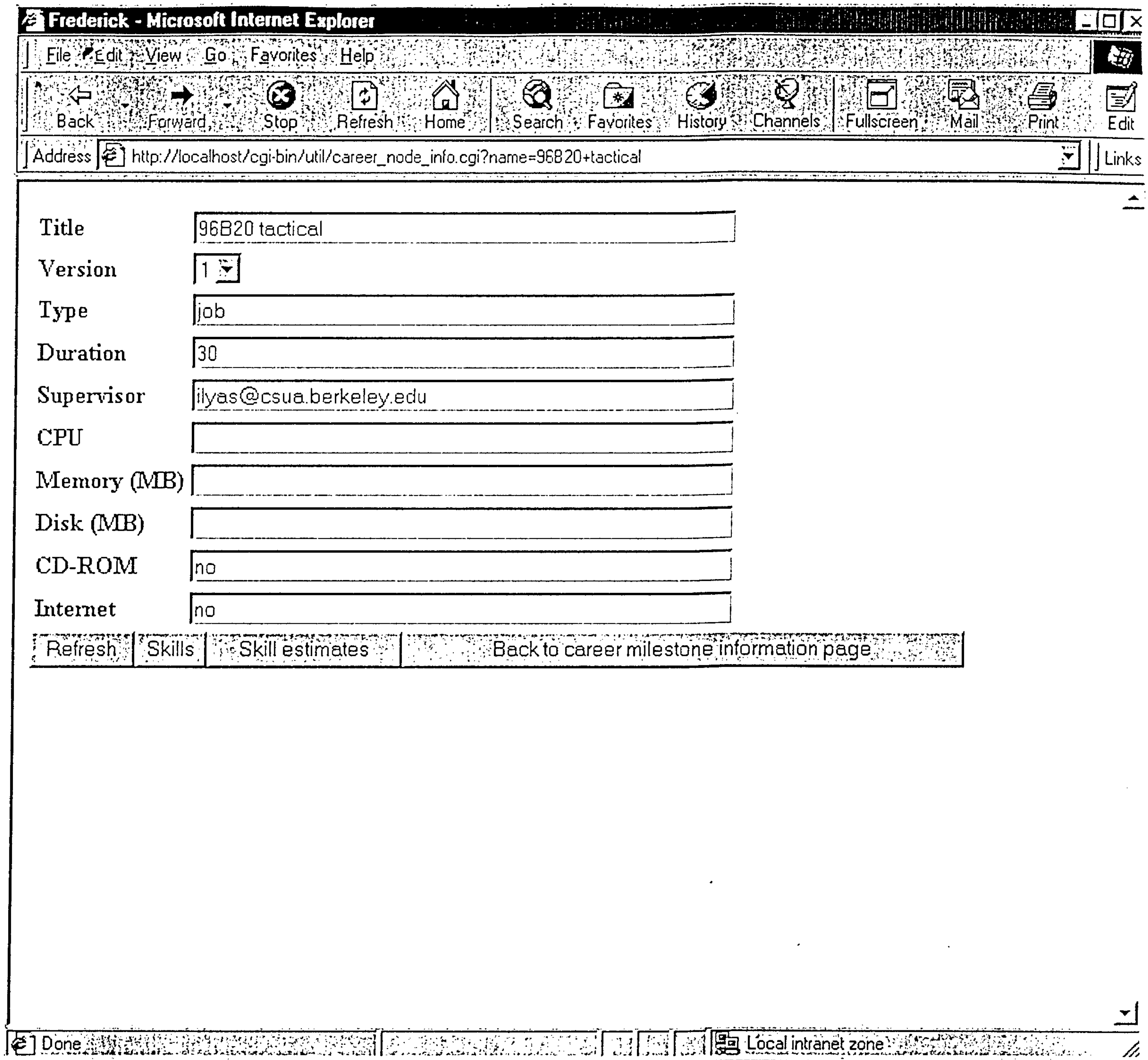
Address http://localhost/cgi-bin/util/view_assigned_skills.cgi?name=stability+and+support+refresher+DL&username=Frederick&version=1 Links

skill chart



Back | Back to career milestone information page

Done | Local intranet zone


Frederick - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Print Edit
Address http://localhost/cgi-bin/util/career_node_info.cgi?name=96B20+tactical
Links
Title 96B20 tactical
Version 1
Type job
Duration 30
Supervisor ilyas@csua.berkeley.edu
CPU
Memory (MB)
Disk (MB)
CD-ROM no
Internet no
Refresh
Skills
Skill estimates
Back to career milestone information page
Done
Local intranet zone